

**Universität Leipzig**  
**Fakultät für Mathematik und Informatik**  
**(Institut für Informatik)**

Automatische Segmentation anatomischer  
Strukturen an MRT-Daten des menschlichen  
Gehirns mit Active Appearance Modellen

**Diplomarbeit**

Leipzig, Oktober 2005

vorgelegt von  
Hagert, Arnold

Fakultät für Mathematik und Informatik  
Prüfungsamt

*für meine Eltern*

# Danksagung

Ich bedanke mich bei Frau Dr. habil. Gabriele Lohmann für die freundliche Aufnahme in ihre Arbeitsgruppe, Überlassung dieses interessanten Themas, die geduldige Betreuung dieser Arbeit und Beantwortung meiner Fragen.

Bei der gesamten Arbeitsgruppe bedanke ich mich für die sehr angenehme Arbeitsatmosphäre und insbesondere bei Dr. Karsten Müller und Dr. Heiko Temming für die Bereitschaft, bei immer wieder auftretenden kleinen Fragen und Problemen schnell und kompetent Hilfestellung zu geben.

Für das Korrekturlesen und ihre motivierende Art danke ich Anne Reiher.

Zum Schluß danke ich meinen Eltern, die mich während meines ganzen Studiums in jeglicher Art und Weise unterstützten.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Hintergrund . . . . .	2
1.2	Aufbau der Arbeit . . . . .	4
1.3	Anmerkung zur Arbeit . . . . .	4
<b>2</b>	<b>Einführung in die Magnetresonanztomographie</b>	<b>5</b>
2.1	Physikalische Grundlagen	
2.2	Bilderzeugung . . . . .	9
2.3	Bildkontrast . . . . .	11
<b>3</b>	<b>Das Shape-Modell</b>	<b>13</b>
3.1	Landmarken . . . . .	13
3.2	Aligning . . . . .	15
3.2.1	Der zweidimensionale Fall . . . . .	16
3.2.2	Der dreidimensionale Fall . . . . .	17
3.3	Erzeugung eines Shape-Modells . . . . .	18
<b>4</b>	<b>Das Appearance-Modell</b>	<b>22</b>
4.1	Textur-Modellierung	
4.2	Das kombinierte Appearance-Modell . . . . .	24
<b>5</b>	<b>Die Objektsuche</b>	<b>25</b>
5.1	Suche in Shape-Modellen . . . . .	25
5.2	Suche in Appearance-Modellen . . . . .	26
5.2.1	Das Erlernen der Parametervorhersage . . . . .	27
5.2.2	Modellanpassung . . . . .	29
5.2.3	Modelleinschätzung . . . . .	30

<b>6</b>	<b>Implementation</b>	<b>32</b>
<b>7</b>	<b>Ergebnisse</b>	<b>34</b>
<b>8</b>	<b>Diskussion</b>	<b>41</b>
<b>A</b>	<b>Aligning zweier Shapes</b>	<b>42</b>
A.1	Aligning zweier 2D Shapes . . . . .	42
A.2	Aligning zweier 3D Shapes . . . . .	43
<b>B</b>	<b>Interpolation</b>	<b>46</b>
B.1	Zweidimensionale Interpolation mit Thin Plate Splines . . . . .	47
B.1.1	Eindimensionale Thin Plate Splines . . . . .	47
B.1.2	Mehrdimensionale Thin Plate Splines . . . . .	48
B.2	Dreidimensionale affine Interpolation . . . . .	49
<b>C</b>	<b>Volumenbestimmung</b>	<b>52</b>
<b>D</b>	<b>Beschleunigung der Hauptkomponentenanalyse</b>	<b>55</b>
	<b>Literaturverzeichnis</b>	<b>57</b>
	<b>Erklärung</b>	<b>59</b>

# Abbildungsverzeichnis

3.1	Randmarkierung für Ventrikel in 2 Dimensionen: Modelle mit einem Objekt bzw. zwei Objekten . . . . .	14
3.2	Randmarkierung und Triangulation der Oberfläche für ein Ventrikel in 3 Dimensionen . . . . .	15
3.3	Beispiel einer Landmarkenverteilung anhand eines kleinen 2D-Modells des Putamen . . . . .	19
5.1	Mittelwertmodell für eine Ventrikelschicht: Auslenkung des 1. bzw. 2. Parameters um je $\pm 2$ Standardabweichungen . . . . .	30
5.2	Darstellung von Auslenkung und der Vorhersage der 1. Iteration für den stärksten Appearance-Parameter und vier lineare Transformationsparameter: $a_x = s \cos(\theta)$ , $a_y = s \sin(\theta)$ , $x$ -und $y$ -Translation. Alle Darstellungen betreffen ein Ventrikelmodell und das Mittel über 10 Testbilder . . . . .	31
7.1	Ein Beispiel für die Klasseneinteilung der Suchergebnisse . . . . .	34
7.2	Eine Verdeutlichung der Problematik, Gewebegrenzen klar zu definieren, am Beispiel zweier Putamen . . . . .	37
B.1	Darstellung einer Schicht beim Interpolationsvergleich — Verfahren nach Arun und Umeyama — Die Textur eines 3D-Objektes wurde in ein anderes projiziert: globale Anwendung links, lokale Anwendung rechts . . . . .	51
C.1	Volumenbestimmung in 2D . . . . .	53

# 1 Einleitung

Ein wichtiges Anliegen der Bildverarbeitung ist das voll- oder halbautomatische Auffinden von Objekten in Bildern — die Segmentation. Sei es die sicherheitstechnisch motivierte Erkennung von Gesichtern, Iris oder Fingerabdrücken oder die medizinisch relevante Frage nach exakten Positionen von Knochen oder Geweben — man steht oft vor dem Problem, ein Objekt einer definierten Klasse, aber leicht veränderlicher Form, möglichst exakt aus einem Bild herauszufiltern.

Soll ein Mensch ein Objekt erkennen, so findet sich in seinem Gedächtnis eine Vorstellung über das generelle Aussehen des Ziels, welches er an das vorliegende Bild anpaßt. Computergestützte Verfahren beruhen analog dazu auf digitalen Modellen.

Im einfachsten Fall kann man ein Standardobjekt systematisch über ein neues Bild wandern lassen, um die passendste Position zu finden. Gerade in der medizinischen Bildverarbeitung ist die Objekterkennung ein Problemfeld, welches komplexere Lösungen verlangt. Die Strukturen beispielsweise von inneren Organen, Knochen oder Gehirnen weisen von Mensch zu Mensch zum Teil eine recht starke Variabilität auf. Modelle müssen aus Rücksicht darauf ebenfalls variabel sein. Dabei ist es notwendig, sie zum einen weit genug zu fassen, um möglichst alle Objekte einer Klasse erkennen zu können, zum anderen benötigt man eine genügende Spezifität, um keine ungewünschten Objekte zu repräsentieren. Ist ein solches deformierbares Modell gegeben, handelt es sich bei einem Segmentationsprozeß um ein Matchingproblem. Dabei werden, ausgehend von einer Grundannahme des Objektes, Modellparameter durch geeignete Algorithmen so angepaßt, daß eine Modellinstanz so gut wie möglich dem zugrundeliegenden Bild bzw. Bildausschnitt entspricht.

Um zu gewährleisten, daß Modelle vollautomatisch die richtigen Objekte finden, liegt der Gedanke nahe, vorhandenes Wissen in ihre Gestaltung einfließen zu lassen. Ähnlich wie ein Mensch die Dinge der Welt mithilfe seines gesammelten Wissens erkennt, wird einem Modell Datenmaterial zugrundegelegt, das zuvor von menschlichen Experten, z. B. erfahrenen Anatomen, erworben wurde. Solches Vorwissen

über zu erwartende Objekte besteht in ihrer Form, ihrer Farb- bzw. Grauwertverteilung oder z. B. bei Bildern aus medizinischen Diagnosesystemen in ihrer Lage zu anderen Strukturen.

Ein Hauptproblem speziell bei Bildern menschlicher Gehirne ist die Tatsache, daß die Grenzen benachbarter Objekten oft fließend sind. Hier versagen Verfahren schnell, die sich an starken Veränderungen der Grauwerte (Kanten) zwischen zwei Arealen orientieren. Das Thema dieser Arbeit ist es, einen speziellen Modellierungsansatz, der den der Active Appearance Models, zu implementieren und auf seine Eignung zu untersuchen, letzterem Problem zu begegnen, um einzelne Objekte in MRT-Bildern des menschlichen Gehirns zu finden.

### 1.1 Hintergrund

Für die Erzeugung deformierbarer Modelle existieren unterschiedliche Ansätze. Eine Möglichkeit, ein Segmentationsproblem zu lösen, besteht darin, ein speziell darauf zugeschnittenes Modell zu kreieren. So erzeugten beispielsweise Yuille, Cohen und Hallinan [25] 1992 aus parametrischen Kreisen und Kreisbögen Modelle des menschlichen Auges und der Lippenform. Derartige Ansätze haben den Vorteil, sehr präzise auf ihr Ziel einstellbar zu sein, aber der betriebene Aufwand der Modellerstellung muß für jede neue Objektklasse wiederholt werden.

Oft nutzt dynamische Modellierung die Nachbildung physikalischer Prozesse. Die Anpassung eines Prototypen an Bildmerkmale erfolgt hier über das Zusammenwirken verschiedener Energiekomponenten. Als *externe* Energie bezeichnet man ihre Fähigkeit, sich an Bildmerkmale anzupassen, als *interne* Energie ein Beharrungsvermögen, welches der Verformbarkeit Grenzen setzt. Ein Gleichgewicht zwischen diesen Energien zu finden, ist Ziel eines Suchalgorithmus'. Eines der frühesten deformierbaren Modelle dieser Klasse sind Active Contour Models, eingeführt von Kass u. a. [13]. Man bezeichnet sie auch als Snakes und nutzt sie heute beispielsweise in der Ultraschalldiagnostik. Bei ihnen werden Objektkonturen als eine Menge parametrischer Kurven zwischen je zwei festen Endpunkten aufgefaßt. Die externe Energie äußert sich durch eine Affinität zu Kanten, die interne durch eine Steifheit, die beispielsweise durch Rauscheinflüsse hervorgerufene Lücken in den Kanten überspielt. Alternativ zu dieser globalen Modellierungsstrategie gibt es auch Ansätze für lokale Snakes [17]. Dabei werden diese als Punktketten aufgefaßt, die sich den Objektgrenzen anglei-



chen. Snakes sind sehr stark von einer guten Initialisierung abhängig und eignen sich auch gut für halbautomatische Systeme. Eine weitere solche Verfahrensklasse sind die von Cohen [3] eingeführte *Ballon*-Modelle. Sie bestehen aus einem Kontur, die gleich einem Ballon von einer inneren Kraft auseinander- und von einer äußeren zusammengedrückt wird. Die innere Kraft entspricht einem generellen Luftdruck, die äußere wird durch Federn mit einer Ruhelänge von Null, an denen Massen hängen dargestellt. Dabei werden die Massen von Kanteneinflüssen in ihrer Umgebung angezogen. Dreidimensionale Feder-Masse-Modelle entwickelte 2005 L. Dornheim [10].

Ein anderer Modellierungsansatz ist die statistische Methode. So führten 1995 Cootes u. a. [4] Active Shape Models (ASM) ein. Bei ihnen wird ein Modell aus einer statistischen Analyse der Formen von Trainingsobjekten erzeugt. Eine Möglichkeit zur dreidimensionalen Modellierung, basierend auf ASM, veröffentlichten 1997 Kelemen, Székely und Gerig [14]. Eine andere, direkte Erweiterung der ASM sind Active Appearance Models (AAM), eingeführt 1998 von Cootes u. a. [5]. Letztere werden in dieser Arbeit betrachtet. Man geht bei ihnen ebenfalls von Vorwissen in Form eines Trainingsbildersatzes aus, in dem Objekte manuell durch Setzen von Landmarkenpunkten definiert wurden. Diese werden bezüglich ihrer Form und Textur analysiert. Dazu dient das Verfahren der Hauptkomponentenanalyse. Das daraus erzeugte Modell beinhaltet die Variabilität der genutzten Bilder und ist in der Lage, in dem damit gesteckten Rahmen neue Instanzen erzeugen. AAM sind generativ, d. h. eine Suchfunktion paßt das Modell iterativ auf ein Objekt im Bild an. Parallel dazu entwickelten Sclaroff und Isidoro ein sehr ähnliches Verfahren [20]; das sogenannte Active Blobs Model. Im Gegensatz zu AAMs beruht es auf einem einzigen Trainingsbild, das segmentiert und mit einem triangulierten Netz überzogen ist. Eine Anpassung erfolgt durch Verformung des Netzes und Resampling der auf das Netz abgebildeten Textur. Das Ziel dabei ist die Minimierung der Texturunterschiede.

Ein Vorteil der statistischen Modellierung ist es, daß sich der zugrundeliegende Algorithmus auf vielen Feldern nutzen läßt, je nachdem welche Trainingsdaten gegeben werden. Weiter ist mit ihm Expertenwissen speicherbar. So lassen erfahrene Anatomen ihr Wissen über den Aufbau von Organen einfließen. Ein solches Modell ist neben seiner Deformierbarkeit spezifisch genug, nicht über den gegebenen Rahmen hinaus beliebige Objekte zu finden. Die Güte eines solchen Modells und seiner Sucherfolge hängt stark von der Spezifität der Trainingslandmarken ab. Je typischer solche Punkte für ein Objekt sind, desto genauer wird die Suchfunktion arbeiten.

Für amorphe Objekte wie beispielsweise stark formvariable Einzeller ist dieser Ansatz also ungeeignet.

Der von Cootes, Edwards und Taylor [5] vorgestellte Algorithmus für zweidimensionale AAM wurde für diese Arbeit implementiert, anhand von Objekten in MRT-Gehirnbildern untersucht und auf Dreidimensionalität erweitert.

## 1.2 Aufbau der Arbeit

Kapitel 1 umfaßt Einführung und Beschreibung des Hintergrundes dieser Arbeit. In Kapitel 2 wird in die Magnetresonanztomographie eingeführt. Dazu werden die physikalischen Grundlagen beleuchtet und die Bilderzeugung beschrieben. Im 3. Kapitel werden Active Shape Models, eine grundlegende Vorstufe zu den Active Appearance Modellen, erläutert. Das 4. Kapitel widmet sich der Erweiterung zu Active Appearance Models. Im 5. Kapitel wird beschrieben, wie mit diesen Modellen Objekte erkannt werden. Das 6. Kapitel beschäftigt sich mit der Implementation. Im 7. Kapitel werden die ermittelten Ergebnisse dargelegt und in Kapitel 8 werden sie bewertet. Im Anhang befinden sich wichtige mathematische Grundlagen, so das Alignment zweier Shapes, was eine Vorbedingung für die Active Shape Models ist, weiter die Interpolationsalgorithmen als ein unbedingter Bestandteil der Modellierung, ebenso die zwei- und dreidimensionale Volumenbestimmung. Die Beschleunigung der Hauptkomponentenanalyse ist vor allem für die dreidimensionale Modellierung unverzichtbar.

## 1.3 Anmerkung zur Arbeit

Die vorliegende Diplomarbeit entstand am Max-Planck-Institut für kognitive Neuropsychologie in Leipzig in der Arbeitsgruppe Bildverarbeitung unter Betreuung von Frau Dr. habil. Gabriele Lohmann. Die dieser Arbeit zugrundeliegenden Bilddaten stammen aus der Datenbank des MPI.

## 2 Einführung in die Magnetresonanztomographie

Vor der Modellierung und Erkennung von Objekten in MR-Schichtbildern soll dieses Kapitel erläutern, wie diese erzeugt werden. Dazu wird erst in die physikalischen Grundlagen der MR-Signalerzeugung eingeführt und danach erklärt werden, wie man aus den Signalen Bilder erzeugt. Dieses Kapitel gibt nur einen grundlegenden Einblick, basierend auf [11, 15, 19]. Die ganze Komplexität der zugrundeliegenden Physik füllt Bände.

### 2.1 Physikalische Grundlagen

Magnetresonanz, oft auch als Nuklearmagnetresonanz (NMR) bezeichnet, fußt auf der Energieabsorption und -emission von Atomkernen. In der klinischen Anwendung nutzt man nur die Signale, die von Wasserstoffkernen ausgehen. Grund dafür ist die Allgegenwart von Wasserstoff im menschlichen Körper mit verschieden großer Dichte in den Geweben. Der Kern eines solchen Atoms besteht nur aus einem Proton. Darum wird im Folgenden auch nur von Protonen die Rede sein. Prinzipiell tritt NMR auch bei anderen Kernen mit einer ungeraden Gesamtanzahl von Protonen und Neutronen auf.

Mit allen Elementarteilchen inklusive des Atomkerns teilt das positiv geladene Proton eine Grundeigenschaft: Es besitzt einen *Spin*. In einem Modell kann man das so sehen, als würde sich das Proton wie ein Kreisel um die eigene Achse drehen, auch wenn der Spin nicht auf eine mechanische Rotation zurückzuführen ist.

Da das Proton rotiert, besitzt es einen Drehimpuls  $M$  und da es zusätzlich noch elektrisch geladen ist, verhält es sich wie ein winziger Stabmagnet mit Nord- und Südpol und hat also ein magnetisches Moment  $B$ . Die Lage der Rotationsachse ist am Vektor  $B$  erkennbar. Entscheidend für die NMR-Technik ist weiter, daß es durch

Magnetfelder beeinflussbar ist und bei einer Bewegung der magnetischen Achse eine Spannung in einer Empfangsspule induziert wird.

Der Terminus Spin stammt aus der Quantenmechanik und ist mit einer Spin-Quantenzahl  $I$  verbunden. Diese reicht für die Atomkerne der Elemente von 0 bis 7, nimmt nur ganz- und halbzahlige Werte an und verändert sich nie. Elemente mit einem Spin von 0 haben eine geradzahlige Gesamtanzahl von Protonen und Neutronen und erzeugen keine Magnetresonanz. Die Spinquantenzahl von Wasserstoff ist  $I = 1/2$ . Aus  $I$  berechnet sich die Anzahl der möglichen diskreten Energielevel  $E_m$  nach Zeeman für einen in einem externen Magnetfeld platzierten Kern. Dabei nimmt  $m$  in Abständen von Eins Werte zwischen  $-I$  und  $+I$  an, für das Proton gibt es also genau 2 solche Zustände  $m_{-1/2}$  und  $m_{+1/2}$ , als parallele und antiparallele Ausrichtung (des Magneten) bezeichnet. Die Besetzung der beiden Energieniveaus hängt von der Feldstärke und der Temperatur ab. Die parallele Ausrichtung hat die geringere potentielle Energie, weshalb sie der günstigere Zustand ist. Thermische Effekte minimieren diesen Effekt jedoch, so daß bei Körpertemperatur beide Ausrichtungen fast gleich häufig auftreten, wichtig ist aber, daß die parallele Ausrichtung statistisch doch etwas bevorteilt ist.

Liegt kein magnetisches Feld an, so sind die magnetischen Momente der Kerne zufällig im Raum verteilt. Wirkt aber ein äußeres Magnetfeld  $B_0$  auf die Protonen, so richten sich die Spins wie Kompaßnadeln daran aus. Äquivalent zu einem Kreisel, der „schwankt“, wenn auf ihn eine äußere Kraft wirkt, reagieren sie dabei mit einem Ausweichen, der sogenannten Präzessionsbewegung. Gleichzeitig geben sie Energie an die Umgebung, also an das Gitter bzw. die chemischen Bindungen in der Nähe, ab (ein Kreisel wird langsamer, was bei einem Proton nicht möglich ist). Bildlich gesehen beschreiben Nord- und Südpol Kreise um je einen Mittelpunkt, der Vektor zwischen diesen entspricht dem Magnetfeldvektor von  $B_0$ . Die Frequenz dieser Präzession ist die sogenannte Larmorfrequenz

$$\omega_0 = \gamma \cdot B_0. \quad (2.1)$$

Dabei sind  $\omega_0$  die Larmorfrequenz,  $\gamma$  das gyromagnetische Verhältnis, eine Konstante, die für die Kerne jedes Elements einen typischen Wert annimmt (42,56 MHz/T für Wasserstoffatomkerne) und  $B_0$  die Magnetfeldstärke.

Mit der Zeit stabilisiert sich das Spin-System und alle präzidierenden Kleinstmagnete sind entlang des  $B_0$ -Feldes ausgerichtet. Diese Richtung ist im Laborkoordi-

natensystem die  $z$ -Achse. Da die magnetischen Momente der Kerne sich addieren, baut sich durch den leichten Überschuß der Parallelausrichtungen eine Längsmagnetisierung  $M_z$ , ein meßbares Magnetfeld in  $z$ -Richtung auf. Die Stärke dieses Feldes berechnet sich nach der Boltzmann-Statistik mit

$$M_z = \frac{(N/V)\gamma^2\hbar I(I+1)B_0}{3k_B T}. \quad (2.2)$$

Dabei sind  $N$  die Gesamtkernzahl,  $V$  das Volumen,  $\hbar$  die Planck-Konstante,  $k_B$  die Boltzmann-Konstante und  $T$  die Temperatur. Der Unterschied in der Verteilung der parallelen und antiparallelen Ausrichtungen ist abhängig von Temperatur und Magnetfeldstärke. Insbesondere ergibt sich durch ein erhöhtes  $B_0$  somit eine Erhöhung von  $M_z$ .  $B_0$  muß sehr stark sein, damit  $M_z$  überhaupt meßbar wird. NMR-Magnetfeldstärken betragen mit 1-9 Tesla, das zigtausendfache der des Erdmagnetfeldes.

Wie etwa Stimmgabeln oder Gitarrensaiten mit Schallwellen der richtigen Frequenz durch Resonanz zur Schwingung angeregt werden können, kann man die Besetzungszahlen der Spinniveaus durch eine elektromagnetische Welle, deren Frequenz der Larmorfrequenz entspricht, stören. Erzeugt wird diese von einem starken Radiosender (für  $B_0 = 1.5T$  benötigt man eine Frequenz von 63.9 MHz, also etwas unterhalb der Sendefrequenzen von UKW-Radio). Ein mit der passenden Leistung (mehrere kW) und Dauer (wenige Millisekunden) eingestrahelter sogenannter 90°-Impuls erreicht, daß sämtliche Spins in die  $x$ - $y$ -Ebene umgeklappt werden. Sämtliche Protonen fallen in das  $m_{-1/2}$ -Energieniveau.  $B_0$  übt nun „Druck“ aus, um die Spins wieder in die  $z$ -Richtung zu drehen. Darum beginnen diese wieder, um die  $z$ -Achse zu präzidieren. Diese Bewegung der Spins beginnt zum selben Zeitpunkt, nämlich nach Beendigung des Impulses. Die Spins sind alle gleich ausgerichtet und präzidieren also alle synchron. Der magnetische Summenvektor ( $M_{xy}$ ) aller Kernmomente ist somit sehr stark. Er bewegt sich direkt nach der Anregung in der  $x$ - $y$ -Ebene und ist meßbar. Die Messung erfolgt dadurch, daß  $M_{xy}$  durch seine Rotation mit der Larmorfrequenz in einer Empfangsspule eine Wechselspannung induziert — das MR-Signal.

Nachdem nun der Impuls beendet ist, läuft das System wieder in den Zustand zurück, in dem es sich vor dem Impuls befand. Dabei treten zwei voneinander unabhängige, Prozesse auf, die T1-Relaxation (Spin-Gitter-Wechselwirkung), welche die  $M_z$ -Magnetisierung allmählich wieder aufbaut und die T2-Relaxation (Spin-Spin-

Wechselwirkung), welche die Signalstärke reduziert.

Betrachten wir zunächst die T1-Relaxation. Nach dem  $90^\circ$ -Impuls ist die longitudinale Feldkomponente ( $M_z$ ) gleich 0, die transversale ( $M_{xy}$ ) maximal und rotiert in der  $x$ - $y$ -Ebene. Wie schon zu Beginn erläutert, bewegen sich die magnetischen Achsen der Protonen allmählich wieder so, daß sie sich erneut nach der  $z$ -Achse ausrichten, während sie die zugeführte Puls-Energie wieder an das umgebende Bindungsgitter abgeben. Die Frage, warum dies kein spontaner Prozeß ist, läßt sich über Einsteins quantenmechanische Beschreibung der Energieemission von Atomen klären [11]. Danach sind Energieabgaben entweder spontan oder von der Umgebung, also anderen Nuklei und Elektronen, verursacht. Weiter ist die Wahrscheinlichkeit spontaner Emission stark frequenzabhängig, und in dem Fenster des Frequenzspektrums, in dem NMR-Frequenzen liegen (um  $10^6$  Hz), treten sie praktisch nicht mehr auf. Der Umgebungseinfluß besteht darin, daß benachbarte Kernmomente ein schwaches Dipolfeld von einigen Millitesla aufbauen. Dieses wird durch thermische Bewegungen der Atome und Moleküle ständig verändert. Solche Fluktuationen bewirken, wenn sie mit der Larmorfrequenz geschehen, daß Energie vom Spinsystem auf die Molekülbewegungen übertragen wird.

Mit dem allmählichen Zurückklappen baut sich das  $M_z$ -Feld wieder auf. Dies geschieht i. A. nach einem Exponentialgesetz:

$$M_z(t) = M_0 (1 - e^{t/T_1}). \quad (2.3)$$

T1 ist dabei die Zeit, nach der ein Teil von  $1 - e^{-1}$ , das sind ca. 63%, der  $M_0$ -Feldstärke wiederhergestellt ist. T1 beträgt, abhängig von der biologischen Gewebeart, bei einem  $B_0$ -Feld von 1.5T zwischen einer halben und mehreren Sekunden. Die Gewebeabhängigkeit folgt daraus, daß die Energieabgabe von den umgebenden Atomen und Molekülen bestimmt wird. Beispielsweise kann in freiem Wasser die Energie nur schlecht abgegeben werden. T1 beträgt dafür ca. 4 Sekunden, dagegen für an Moleküle „gebundenes“ Wasser nur 0.4 bis 0.8 Sekunden.

Der zweite Prozeß, die T2-Relaxation, baut das Magnetfeld  $M_z$  und damit das MR-Signal ab. Er beruht auf der Desynchronisierung bzw. Dephasierung der das rotierende  $M_{xy}$ -Feld bildenden Protonen. Als Phase bezeichnet man hier einen Positionsunterschied gleich schnell präzidierender Spins, vergleichbar mit zwei Uhren, von denen eine ein paar Minuten vor- bzw. nachgeht. Direkt nach der Anregung rotieren alle Spins synchron, sie sind „in Phase“, d. h. sie haben zur selben Zeit die

gleiche räumliche Ausrichtung. Die Summe ihrer magnetischen Momente ist maximal. Aus zwei Gründen, die später dargelegt werden, geschieht es, daß die Spins nach und nach „außer Phase“ geraten — das  $M_{xy}$ -Feld wird geringer (z.B. heben sich zwei Spins mit einer Phase von  $180^\circ$  vollständig auf). Bei einer statistischen Gleichverteilung der Phasen ist die transversale Magnetisierung dann gleich 0. Die Zeit  $T_2$  beträgt lediglich 100-300ms, so daß das Signal bereits deutlich eher verschwunden ist, als die Längsmagnetisierung  $M_z$  gemäß (2.3) wieder aufgebaut ist. Auch dieser Prozeß genügt für Flüssigkeiten meist einem Exponentialgesetz:

$$M_{xy} = M_0 e^{-t/T_2}. \quad (2.4)$$

$T_2$  ist die Zeit, die benötigt wird, bis ein Teil von  $e^{-1}$  (ca. 37%) des  $M_{xy}$ -Feldes zerfallen ist und ist generell nicht beeinflussbar.

Die erste der beiden erwähnten Ursachen, die zur Dephasierung beitragen, ist die reine Spin-Spin-Wechselwirkung. Die Spins beeinflussen sich durch ihre eigenen winzigen Magnetfelder gegenseitig. Dadurch kommt es zu permanenten kleinen lokalen Veränderungen im Magnetfeld. An diesen Stellen präzidieren die Spins dann ein wenig schneller oder langsamer, weil die Präzessionsfrequenz von der Magnetfeldstärke abhängig ist. So fällt das System allmählich in eine zufällige Verteilung der Phasen zurück. Der zweite Grund ergibt sich aus systemimmanenten Magnetfeldinhomogenitäten. Verursacher sind das Gerät und das Untersuchungsobjekt selbst. Die Dephasierungszeit wird durch diesen Extraeffekt zu einer Zeit  $T_2^*$  verkürzt. Diesem zweiten Effekt kann begegnet werden, indem man zu einer bestimmten Zeit  $t$  einen  $180^\circ$ -Impuls sendet. Dieser invertiert die Reihenfolge der Spins so, daß diejenigen, die aufgrund der äußeren Magnetfeldinhomogenität schneller präzidierten, nun „hinterherlaufen“ und die anderen wieder einholen und umgekehrt. Nach der doppelten Zeit  $2t$  ist dieser Effekt ausgeglichen. Man erhält ein Spin-Echo. Durch dieses Mittel verläuft die Dephasierung wieder mit  $T_2$ .

## 2.2 Bilderzeugung

Um ein Bild des Untersuchungsobjektes zu erzeugen, benötigt man Informationen, welches Signal von welchem Raumpunkt stammt. Der Begriff Raumpunkt ist nicht wörtlich zu nehmen; betrachtet werden Voxel, das sind kleine Würfel, die durch ein diskretes Raster durch das Objekt entstehen. Die  $z$ -Koordinate wird dadurch fest-

gelegt, daß transversale (axiale) Schichten untersucht werden. Zur Erinnerung, das Magnetfeld  $B_0$  verläuft in  $z$ -Richtung. Das ist auch die Richtung, in der bei einer Tomographie der Patient liegt. Um nicht das komplette Objekt, sondern nur eine einzelne Schicht zur Signalabgabe anzuregen, wird das Magnetfeld  $B_0$  inhomogen gemacht. Eine zusätzliche Magnetspule verstärkt  $B_0$  am Kopfende und verringert es am Fußende. Man sagt,  $B_0$  hat einen Gradienten entlang der  $z$ -Achse. Dadurch ergibt sich eine Veränderung der Larmorfrequenzen der Spins, da diese direkt proportional von der Magnetfeldstärke abhängen. Am Fußende finden sich also geringere Larmorfrequenzen als am Kopfende. Der  $90^\circ$ -Impuls wirkt aufgrund der Resonanzbedingung nur auf Spins einer bestimmten Larmorfrequenz bzw. eines kleinen Frequenzbereiches, wodurch sich mit seiner Wahl bestimmen läßt, welche Schicht angeregt wird.

Zu ermitteln bleibt noch, welches Voxel der Schicht welchen Anteil des gemessenen Signals hat. Die  $x$ -Koordinate läßt sich direkt ermitteln. Dazu wird eine Frequenzkodierung durchgeführt. Es wird durch eine weitere Magnetspule ein  $x$ -Gradient eingeschaltet, so daß das Magnetfeld von links nach rechts stärker wird. Dadurch präzidieren die Spins links langsamer als rechts. Die Messung des MR-Signals ergibt nun nicht mehr die einzelne Larmorfrequenz, sondern ein Spektrum verschiedener Frequenzen. Tiefe Frequenzen stammen dabei vom linken Rand, hohe vom rechten. Eine Fouriertransformation ist in der Lage, aus dem MR-Signal die einzelnen Frequenzen zu extrahieren, wodurch die  $x$ -Koordinaten der Voxel innerhalb der Schicht definiert werden.

Zur Ermittlung der  $y$ -Koordinate nutzt man eine Phasenkodierung. Schaltet man einen  $y$ -Gradienten so ein, daß im oberen Bereich das Magnetfeld gestärkt und im unteren geschwächt wird, so hat dies den Effekt, daß die Larmorfrequenzen im oberen Bereich etwas höher sind als im unteren. Angeregte Spins mit einer höheren Larmorfrequenz präzidieren etwas schneller und erreichen damit einen Phasenvorsprung gegenüber den Spins im unteren Bereich. Nach Abschaltung des Gradienten ist die Präzessionsgeschwindigkeit wieder gleich, aber man hat eine Phasenaufteilung erreicht. Je weiter oben ein Spin sich befindet, desto größer ist sein Phasenvorsprung. Dadurch ist jede Zeile der Schicht eindeutig definiert.

Leider ist die Phase von Spins nicht direkt meßbar. Die Messung einer Schicht kann als eine Gleichung mit bekanntem Ergebnis, aber vielen Unbekannten, den  $y$ -Koordinaten, aufgefaßt werden. Um diese zu ermitteln, benötigt man ein (lineares) Gleichungssystem von so vielen unabhängigen Gleichungen, wie Unbekannte zu fin-



den sind. Dafür werden in einer Schicht genau so viele Anregungen und Messungen durchgeführt, jedes mal mit einer anderen, stärkeren Phasenkodierung. Der Zusammenhang zwischen den verschiedenen Stärken und Ergebnissen liefert über eine weitere Fouriertransformation die Phasenaufteilung. Insgesamt ist also für jede Schicht eine zweidimensionale Fouriertransformation durchzuführen, wodurch sich auch der Name der Methode herleitet: 2DFT-Rekonstruktion. Die 2DFT ist eine sehr aufwendige Berechnung, für die das Gerät einen eigens darauf spezialisierten Rechner, den Array-Prozessor, besitzt.

Es gibt auch die Möglichkeit, anstelle der Schichtselektion das ganze Volumen auf einmal anzuregen und durch einen weiteren Gradienten die  $z$ -Ortsinformation zu ermitteln. Dazu benötigt man eine dreidimensionale Fouriertransformation. Ergebnis einer solchen Messung ist ein kompletter Volumendatensatz mit sehr großer Auflösung. Dieses Verfahren wird daher z. B. angewandt, um dünne Schichten optisch aufzulösen. Das 2DFT-Signal einer solchen ist sehr schwach, da weniger signalgebende Spins vorhanden sind. Dadurch kann es durch das Hintergrundrauschen überdeckt werden. Der Nachteil besteht in der deutlich längeren Aufnahmezeit im Vergleich zu einer Schichtmessung.

## 2.3 Bildkontrast

Es gibt drei gewebetypische Parameter, welche die Signalstärke, also die Helligkeit der Voxel im Bild bestimmen. Diese sind die Protonendichte, also der Wasseranteil des Gewebes, die T1- und die T2-Zeit. Je nach Art der Messung kann man den Einfluß dieser Parameter verschieden gewichten und dadurch gezielt Einfluß auf den optisch erkennbaren Unterschied zwischen bestimmten Gewebetypen nehmen.

T1 ist ein Maß dafür, wie schnell sich die Spins nach einer Anregung wieder in  $z$ -Richtung ausrichten. Je stärker diese Ausrichtung fortgeschritten ist, desto stärker ist die Längsmagnetisierung  $M_z$ . Je stärker  $M_z$ , desto stärker ist das Signal bei der nächsten Anregung, weil die Richtungsänderung beim Umklappen  $90^\circ$ -Impuls) größer ist. Da eine Schicht für eine Messung viele Male angeregt wird, ergibt sich bei einer kurzen Repetitionszeit (unter 600ms) der Effekt, daß Gewebe mit langsamer T1-Relaxation bei der nächsten Anregung ein schwächeres Signal abgeben als Gewebe, die schneller T1-relaxieren. Damit erscheinen Gewebe mit kleiner T1, wie z. B. Fett, im Bild heller als Gewebe mit großer T1, wie z. B. Muskeln.

Eine T2-Gewichtung bestimmt man durch die Wahl von Echozeiten. Wie am Ende des physikalischen Grundlagenkapitels beschrieben, lassen sich die systematisch bedingten dephasierungsfördernden Einflüsse durch einen  $180^\circ$ -Impuls nach einer Zeit  $t$  umkehren, so daß nach einer gewissen Zeit  $2t$  das Signal wiederhergestellt ist. Diese Zeit ist die Echozeit, zu der dann die Signalmessung ausgeführt wird. T2-Zeiten liegen in einem Bereich von einigen hundert Millisekunden. Wählt man eine sehr kurze Echozeit, so hat die T2-Relaxation gerade erst begonnen und das Signal aller Gewebe ist noch ähnlich stark. Liegt die Echozeit aber in einem Bereich, in dem die T2-Relaxation schon weit fortgeschritten ist, so ergeben Gewebe mit einer langen T2 noch viel Signal und erscheinen also hell. Gewebe mit kurzer T2 dagegen sind schon weitgehend relaxiert und erscheinen also dunkel.

Natürlich sind beide Gewichtungen nicht voneinander unabhängig. So erhält man eine T1-Gewichtung nur, wenn die Repetitionszeit und die Echozeit kurz sind. Eine T2-Gewichtung ergibt sich, wenn beide Zeiten lang gewählt werden. Ist die Echozeit kurz und die Repetitionszeit lang, so werden beide Effekte minimiert und man erhält ein Bild, daß nach der Protonendichte gewichtet ist.

Bedeutende Unterschiede zwischen T1- und T2-Gewichtung ergeben sich beispielsweise bei Entzündungen oder Tumoren, die bei T1-Gewichtung dunkel und bei T2-Gewichtung hell erscheinen.

Im Vergleich zur aufgewandten Impulsleistung von bis zu 20kW ist die Leistung des Signals mit einigen  $\mu\text{W}$  sehr schwach. Damit es nicht schwächer als das Rauschen ist, entsteht eine Begrenzung in der Bildgebung. So ergibt eine höhere Auflösung ein schwächeres Signal, da die Anzahl der Voxel, auf die es sich verteilt, steigt. Dünnere Schichten ergeben aufgrund der geringeren Signalintensität einen geringeren Abstand vom Rauschen. Weitere solche negativen Faktoren sind kurze Repetitions- und lange Echozeiten. Dem Rauscheinfluß kann man begegnen, in dem man  $n$  Messungen vollzieht und die Signale aufsummiert. Das echte Signal erhöht sich um den entsprechenden Faktor  $n$ , das Rauschen als statistischer Wert nur um  $\sqrt{n}$ . Natürlich steigt die benötigte Zeit auf das  $n$ -fache. Auch eine erweiterte Empfangstechnik ermöglicht detailreichere Untersuchungen. Am einfachsten ist die Messung mit der selben Spule, die auch die Anregung erzeugt. Besser ist die Wahl einer oder mehrerer zusätzlicher Oberflächenspulen.

## 3 Das Shape-Modell

Wiewohl sie als eigenständige Modelle zu sehen sind, bilden Shape-Modelle die Grundlage oder eine Vorstufe zu den Appearance-Modellen. Ein Shape ist die Beschreibung der Form eines Objektes durch eine geordnete Menge von  $n$  Randpunkten (Landmarken). Diese können, wie im weiteren beschrieben, ganz einfach zwei- oder dreidimensional sein, aber auch weitere Dimensionen wie eine zeitliche sind möglich.

Ein Shape-Modell wird aufgrund eines Trainingsdatensatzes ähnlicher Objekte erzeugt, in welchem die Shapes zu den Objekten manuell ermittelt wurden. Dieser Trainingsdatensatz wird einer Analyse der Variation innerhalb der Shapes unterzogen. Die daraus resultierenden Parameter beschreiben das Shape-Modell und ermöglichen es, neue Shapes dieser Objektklasse zu erzeugen.

### 3.1 Landmarken

Die Güte eines Shape-Modells hängt sehr stark von einer guten Wahl der Landmarken im Trainingsdatensatz ab. Es ist nötig, eine Menge von Bildern mit einem Set von korrespondierenden Punkten zu versehen. Wenn also z. B. im ersten zweidimensionalen Bild  $\mathbf{B}_0$  der Punkt  $(x_{00}, y_{00})$  die obere linke Ecke eines Objektes bezeichnet, dann muß in jedem anderen Bild  $\mathbf{B}_i$  der Punkt  $(x_{i0}, y_{i0})$  ebenfalls die obere linke Ecke des ähnlichen Objektes bezeichnen.

Dieser Markierungsprozeß ist mithin ein sehr langwieriger, besonders was das Bearbeiten von 3D-Objekten anbetrifft. Darum beschäftigen sich die während dieser Arbeit erzeugten Modelle der Einfachheit halber mit recht klar zu erkennenden Objekten, vornehmlich den Ventrikeln. Gerade für MR-Gehirnbilder wird es aber besonders für den praktischen Einsatz unumgänglich sein, einen spezialisierten Anatomen zu befragen, um brauchbare Landmarken zu erhalten, denn ein Großteil der Grenzen zwischen den verschiedenen Hirngeweben ist für Laien optisch nur schwer auszumachen.

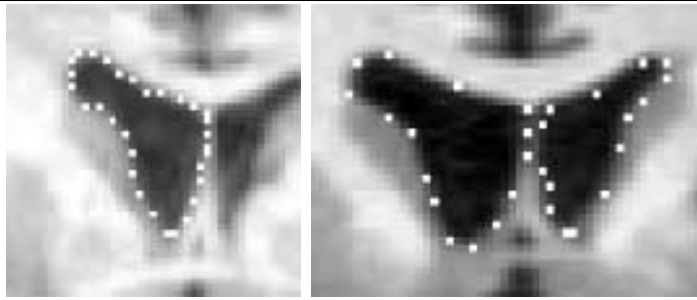


Abbildung 3.1: Randmarkierung für Ventrikel in 2 Dimensionen: Modelle mit einem Objekt bzw. zwei Objekten

---

Im zweidimensionalen Fall setzt man zunächst Landmarken auf klar erkennbare Eckpunkte des Objektes. Da diese in der Regel nicht ausreichen, um das Objekt zu gut rekonstruieren, fügt man in regelmäßigen Abständen weitere Punkte entlang der Objektgrenze ein. Diese, mit Geraden verbunden, müssen dann die Objektform gut wiedergeben, wie in Abbildung (3.1) dargestellt ist. Bei dieser Darstellung sind nicht etwa die ganzen Pixel die Landmarken, sondern nur der Punkt, der ihre obere linke Ecke bezeichnet.

Im dreidimensionalen Fall ist das Finden von Landmarken deutlich schwieriger. Ein Bild liegt als Quader vor, durch den man sich schichtweise in alle Richtungen bewegen kann, eine 3D-Ansicht einzelner Teile ist damit naturgemäß zunächst nicht möglich. Daher wurde für die Modelle innerhalb dieser Arbeit ein Objekt aus einem MR-Bild zunächst schichtweise mit vorläufigen Landmarken versehen. Eine aus diesen erzeugte frei drehbare 3D-Ansicht ermöglichte die Platzierung von Landmarken an hervorstechenden Positionen. Zwischen diesen wurden möglichst regelmäßig weitere Punkte eingefügt. Eine manuell ermittelte Triangulation der Oberflächenpunkte ermöglicht dann die in Abbildung Objektrekonstruktion (C). Ein Beispiel ist in Abbildung (3.2) wiedergegeben. Diese Triangulation muß nur einmal durchgeführt werden und gilt dann für alle Modellinstanzen. Dennoch handelt es sich dabei um einen recht aufwendigen Prozeß. Zudem verhindert jeder Fehler (Öffnungen in der Oberfläche) die Entscheidung, ob sich ein Punkt inner- oder außerhalb der Oberfläche befindet. Der Gedanke an eine automatische Triangulation sollte bei weiterer Beschäftigung mit dreidimensionalen Active-Appearance-Modellen nicht außer acht gelassen werden.

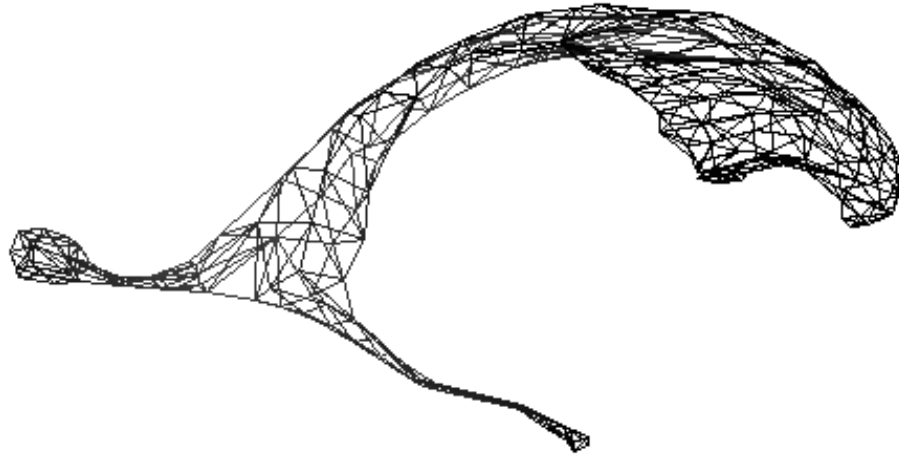


Abbildung 3.2: Randmarkierung und Triangulation der Oberfläche für ein Ventrikel in 3 Dimensionen

---

Ein Shape aus  $n$  solchen Landmarken ist nun z. B. im Zweidimensionalen der Vektor

$$\mathbf{x} = (x_0, y_0, x_1, y_1, \dots, x_{n-1}, y_{n-1})^T. \quad (3.1)$$

Ein Trainingsdatensatz mit  $k$  Bildern besteht aus  $k$  solchen Vektoren  $\mathbf{x}_i$ . Vor der statistischen Analyse dieser Trainingsdaten ist es zur besseren Vergleichbarkeit noch wichtig, durch globale Transformationen (Translation, Rotation und Skalierung), welche die Form der Shapes nicht verändern, Variation zu entfernen, die für alle Punkte eines Shapes gleich wäre. Dieses geschieht während des Alignings.

## 3.2 Aligning

Ziel des Alignings ist es, die Vergleichbarkeit der Trainings-Shapes zu verbessern, indem sie durch globale Transformationen einander möglichst optimal angenähert werden. Eine verbreitete Methode dafür ist die Procrustes-Analyse [1]. Als mögliche Operationen auf den Shapes wurden Rotation, Translation und Skalierung verwendet.

### 3.2.1 Der zweidimensionale Fall

Der Vektor  $\mathbf{x}_i$  beschreibt die  $n$  Punkte im  $i$ ten Shape mit

$$\mathbf{x}_i = (x_{i0}, y_{i0}, x_{i1}, y_{i1}, \dots, x_{in-1}, y_{in-1})^T. \quad (3.2)$$

Die zu tätigen globalen Transformationen Translation um  $(t_x, t_y)$ , Rotation um  $\theta$  und Skalierung um  $s$  sind zusammengefaßt in  $\mathbf{T}(s, \theta, t_x, t_y)$ . Sei das Problem zunächst darauf reduziert, lediglich zwei Shapes  $\mathbf{x}_i$  und  $\mathbf{x}_j$  miteinander zu alignen, so lassen sich Parameter  $s_j$ ,  $\theta_j$ ,  $t_{xj}$  und  $t_{yj}$  finden, mit denen  $\mathbf{T}(s_j, \theta_j, t_{xj}, t_{yj})[\mathbf{x}_j]$  und  $\mathbf{x}_i$  den gewichtete quadratischen Fehler

$$E = (\mathbf{T}(s_j, \theta_j, t_{xj}, t_{yj})[\mathbf{x}_j] - \mathbf{x}_i)^T \mathbf{W} (\mathbf{T}(s_j, \theta_j, t_{xj}, t_{yj})[\mathbf{x}_j] - \mathbf{x}_i) \quad (3.3)$$

minimieren. Dabei ist

$$\mathbf{T}(s_j, \theta_j, t_{xj}, t_{yj}) \begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{pmatrix} (s_j \cos \theta_j) x_{jk} - (s_j \sin \theta_j) y_{jk} + t_{xj} \\ (s_j \sin \theta_j) x_{jk} + (s_j \cos \theta_j) y_{jk} + t_{yj} \end{pmatrix}, \quad (3.4)$$

und bei  $\mathbf{W}$  handelt es sich um eine Diagonalmatrix mit Gewichten für jeden Punkt. Gewichtet wird, um Punkten, die über den kompletten Trainingsdatensatz stabiler als andere erscheinen, Priorität bei der Parameterbestimmung zu verleihen. Als mögliches Kriterium für die Gewichtung eines Punktes gilt hier, wie Cootes u. a. ([4], S. 42) vorschlagen, die Variation seines Abstandes zu allen anderen Punkten. Seien also  $\mathbf{R}_{kl}$  der Abstand zwischen den Punkten  $k$  und  $l$  innerhalb eines Shapes und  $\mathbf{V}_{\mathbf{R}_{kl}}$  die Varianz dieses Abstandes über alle Shapes gesehen. Damit ergibt sich als Gewicht für einen Punkt  $k$

$$w_k = \left( \sum_{l=0}^n \mathbf{V}_{\mathbf{R}_{kl}} \right)^{-1}. \quad (3.5)$$

Wenn also ein Punkt  $k$ , über alle Shapes gesehen, seinen Abstand zu anderen Punkten  $l$  stark verändert, wird die Summe der Varianzen  $\mathbf{V}_{\mathbf{R}_{kl}}$  groß werden und damit das Gewicht klein und umgekehrt.

Für die Berechnung setzt man  $a_{xj} = s_j \cos \theta_j$  und  $a_{yj} = s_j \sin \theta_j$  — Details der Parameterbestimmung von  $(a_x, a_y, t_x, t_y)$  finden sich im Anhang A (S. 42).

Um nun alle  $N$  Shapes des Trainingsdatensatzes optimal zu alignen, wird der

folgende Algorithmus (aus [4]) genutzt:

1. Verschiebe jedes Shape so, daß sein Schwerpunkt im Koordinatenursprung liegt.
2. Wähle ein beliebiges Shape als erste Schätzung für den Mittelwert  $\bar{\mathbf{x}}_0$ .
3. Aligne jedes Shape mit  $\bar{\mathbf{x}}_0$  durch Translation, Rotation und Skalierung.
4. Berechne ein Mittelwertshape  $\bar{\mathbf{x}}_1$  aus den aligneten Shapes.
5. Normalisiere Skalierung und Schwerpunkt von  $\bar{\mathbf{x}}_1$  und die Ausrichtung durch Aligning mit  $\bar{\mathbf{x}}_0$ .
6. Wenn der Prozeß noch nicht konvergiert ist, so setze  $\bar{\mathbf{x}}_0 = \bar{\mathbf{x}}_1$  und gehe zu 3.

Welches Shape man als erste Schätzung einsetzt, ist nicht entscheidend; die Ergebnisse diesbezüglicher Experimente unterschieden sich nur marginal.

Als Konvergenzkriterium für Punkt 6 berechnet man beispielsweise die Summe der Abweichungen äquivalenter Punkte zwischen dem vorigen und dem aktuellen Mittelwertshape. Wenn sich diese nicht mehr stärker als um einen gesetzten Wert ändert, so wird Konvergenz angenommen.

#### 3.2.2 Der dreidimensionale Fall

Im Unterschied zum 2D-Alignment wird in diesem Rahmen auf die Optimierung der räumlichen Ausrichtung, also auf die Rotation verzichtet. Generell sollte dies nicht getan werden, hier ist es aber möglich, da die zur Verfügung stehenden Bilder bereits in das Talairach-System [9] eingeordnet wurden, wodurch eine sehr ähnliche räumliche Orientierung der untersuchten Objekte garantiert ist. Dieses Weglassen erspart vor allem für die Suche Rechenzeit. Für Untersuchungen ohne dieses Nebenbedingung ist es kein Problem, die Rotation mitzubetrachten. Der Arunsche Algorithmus zum Matchen zweier 3D-Punktwolken wurde für 3D-Interpolationsprobleme im Rahmen dieser Arbeit schon implementiert (B.2, S. 49). Alternativ wäre die Anwendung eines iterative closest point (ICP) - Algorithmus' [16, 23] denkbar. Das Hinzufügen weiterer zu optimierender Parameter zur implementierten Lösung ist leicht möglich.

Das hier angewandte Verfahren ist, von der genannten Vereinfachung abgesehen, identisch zur zweidimensionalen Variante, der quadratische Fehler ist nun

$$E = (\mathbf{T}(s_j, t_{xj}, t_{yj}, t_{zj}) [\mathbf{x}_j] - \mathbf{x}_i)^T \mathbf{W} (\mathbf{T}(s_j, t_{xj}, t_{yj}, t_{zj}) [\mathbf{x}_j] - \mathbf{x}_i), \quad (3.6)$$

mit

$$\mathbf{T}(s_j, t_{xj}, t_{yj}, t_{zj}) \begin{bmatrix} x_{jk} \\ y_{jk} \\ z_{jk} \end{bmatrix} = \begin{pmatrix} s_j x_{jk} + t_{xj} \\ s_j y_{jk} + t_{yj} \\ s_j z_{jk} + t_{zj} \end{pmatrix}. \quad (3.7)$$

Details zur Bestimmung der Parameter  $s_j$ ,  $t_{xj}$ ,  $t_{yj}$  und  $t_{zj}$  finden sich im Anhang A (S. 42).

### 3.3 Erzeugung eines Shape-Modells

Ziel dieses Kapitels ist es, zu beschreiben, wie aus einem aligneten Trainingsdatensatz zunächst ein Shape-Modell entsteht. Gegeben sei also eine Menge ähnlicher Shapes  $\mathbf{x}_i$ , alle mit Schwerpunkt im Koordinatenursprung. Wie in Abb. 3.3 verdeutlicht, bilden die zueinander äquivalenten Punkte mehr oder weniger dichte Wolken um einen jeweiligen Mittelwert. Gewünscht ist nun eine Beschreibung der Variation dieser Punkte, genauer ein Modell der Form  $\mathbf{x} = M(\bar{\mathbf{x}}, \mathbf{p})$  mit  $M$  als Modellfunktion und  $\mathbf{p}$  als einem Parametervektor. Wichtig ist weiter, daß die einzelnen Punkte durchaus nicht voneinander unabhängig sind, sondern auch untereinander korrelieren.

Zur Modellierung wird das Verfahren der Hauptachsentransformation (Principal Component Analysis, PCA, [21]) angewandt. Jedes alignete Shape  $\mathbf{x}_i$  ist zunächst ein  $n \cdot \dim$ -dimensionaler Vektor, kann aber auch als ein einzelner Punkt im  $n \cdot \dim$ -dimensionalen Raum angesehen werden. Diese wiederum bilden eine Punktwolke in jenem hochdimensionalen Raum. Da alle Shapes einander relativ ähnlich sind, liegen diese auch wirklich nur innerhalb eines gewissen, recht begrenzten Raumbereiches und verteilen sich nicht weitläufig im gesamten  $n \cdot \dim$ -Raum. Darum wird nun angenommen, diese Wolke entspräche ungefähr einem Ellipsoiden, der eben diesen erlaubten Bereich beschreibt. Dessen Hauptachsen zu bestimmen, ist nun Aufgabe der PCA. Diese bilden eine Basis des  $n \cdot \dim$ -Raumes — sind sie bekannt, so kann man jeden Punkt des Ellipsoiden durch Linearkombination der Achsenvektoren erreichen, wodurch auch schon das Modellierungsprinzip ersichtlich wird, denn jeder





Abbildung 3.3: Beispiel einer Landmarkenverteilung anhand eines kleinen 2D-Modells des Putamen

---

so erreichbare Punkt stellt ja wieder ein Shape dar, das den im Trainingsdatensatz enthaltenen ähnlich ist, solange die Parameter keine extremen Werte annehmen. Die Anzahl der Parameter entspricht der Anzahl der Achsen, beträgt also  $n \cdot \dim$ . Diese Zahl kann vor allem bei der Modellierung dreidimensionaler Objekte sehr groß werden, was den Rechenaufwand stark erhöht. Die PCA liefert zugleich eine einfache Möglichkeit, dieses Problem zu reduzieren, doch dazu mehr nach der Beschreibung der zugrundeliegenden Mathematik:

1. benötigt man den Mittelwert der Daten ( $N$  Shapes mit je  $n$  Punkten der Dimension  $d$ )

$$\bar{\mathbf{x}} = \begin{pmatrix} \bar{x}_0 \\ \vdots \\ \bar{x}_{d-1} \end{pmatrix} = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{x}_i = \frac{1}{N} \sum_{i=0}^{N-1} \begin{pmatrix} x_{i0} \\ \vdots \\ x_{id-1} \end{pmatrix}, \quad (3.8)$$

2. die Kovarianzmatrix

$$\mathbf{S} = \frac{1}{N} \sum_{i=0}^{N-1} (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T \quad \text{und} \quad (3.9)$$

3. die Eigenvektoren  $\phi_i$  und die korrespondierenden Eigenwerte  $\lambda_i$  zu  $\mathbf{S}$  nach

$$\mathbf{S}\phi_i = \lambda_i\phi_i. \quad (3.10)$$

Nun beschreiben die Eigenvektoren, zu denen die größten Eigenwerte gehören, auch die längsten Achsen des Ellipsoides. Genauer bilden sie sogar eine Orthonormalbasis des  $n \cdot \dim$ -Raumes, d.h. sie stehen senkrecht aufeinander und sind damit linear unabhängig ( $\forall i, j : \vec{e}_i \cdot \vec{e}_j = 0$ ) und sind normiert 1 ( $\forall i : \|\vec{e}_i\| = 1$ ). Sei also  $\Phi$  der Vektor, der die Eigenvektoren, sortiert nach der Größe der zugehörigen Eigenwerte, enthält. Das gesuchte komplette Modell ergibt sich damit zu

$$\mathbf{x} = \bar{\mathbf{x}} + \Phi \mathbf{p}. \quad (3.11)$$

Jeder Parameter  $p_i$  repräsentiert die Varianz innerhalb einer der Hauptachsen.

Man hätte prinzipiell auch statt  $\Phi$  einfach die kanonische Orthonormalbasis setzen können. Allerdings ergibt sich aus der Art der Bestimmung von  $\Phi$  ein mächtiger Vorteil. Das aufgrund seiner teilweise enormen Größe etwas unhandliche Modell soll reduziert werden. Da Eigenvektoren (Basisvektoren) zu sehr kleinen Eigenwerten von  $\mathbf{S}$  nur einen sehr kleinen Teil der Gesamtvarianz erklären, aber in ihrer Anzahl gerade bei großen Modellen stark überwiegen, ignoriert man einfach einen Teil der Eigenvektor/Eigenwert-Paare. Der Verlust wiegt den Geschwindigkeitsvorteil mehr als auf. Dazu berechnet man die totale Varianz aller Variablen als

$$V_T = \sum_{i=0}^{n \cdot d - 1} \lambda_i \quad (3.12)$$

und wählt die  $t$  größten Eigenwerte so, daß

$$\sum_{i=0}^{t-1} \lambda_i \geq m V_T. \quad (3.13)$$

Typischerweise setzt man für  $m$  Werte  $\geq 0.95$  ( $m \leq 1$ ). Bei Versuchen lieferten aber auch Werte über 0.8 nur wenig schlechtere Ergebnisse. Nun sei der Vektor  $\Phi_s = (\phi_0, \dots, \phi_{t-1})$  der Vektor, der die Eigenvektoren der ersten  $t$  Eigenwerte enthält und  $\mathbf{b}_s = (p_0, \dots, p_{t-1})$  ein Gewichtungsvektor. Dadurch ergibt sich das approximierte

Modell letztlich zu

$$\mathbf{x} \approx \bar{\mathbf{x}} + \Phi_s \mathbf{b}_s. \quad (3.14)$$

Die Gewichte in  $\mathbf{b}_s$  sind die Parameter des verformbaren Shape-Modells und durch ihre Veränderung lassen sich neue Objekte erzeugen. Durch Einschränkung in der Variation der Parameter läßt sich vermeiden, daß extrem verzerrte Shapes entstehen. Eine typische Grenze für den Parameter  $b_i$  mit der Varianz  $\lambda_i$  ist  $\pm 3\sqrt{\lambda_i}$ , eine statistische Standardgröße.

## 4 Das Appearance-Modell

Zur Modellierung eines Objektes gehört nicht nur die Modellierung seiner äußeren Form wie im Shape-Modell, sondern auch die seiner Textur, hier der Grauwerte aller enthaltenen Punkte. Dieses Kapitel widmet sich daher der Modellierung von Texturen und der Korrelationen zwischen ihnen und den Objektshapes. Bei einer Textur handele es sich im Folgenden um einen reinen Grauwertvektor  $\mathbf{g}$ , der durch Abrastern des Objektes nach einem festen Muster ermittelt bzw. wieder zugeordnet wird.

### 4.1 Textur-Modellierung

Gegeben sei, wie im vorigen Kapitel, eine Menge Bilder, in denen Objekte von Hand markiert wurden. Der Aligning-Prozeß sei abgeschlossen, so daß ein Mittelwertshape vorliege. Um die Objekttexturen vergleichbar zu machen und die durch verschiedene Objektformen hervorgerufene Texturvarianz zu eliminieren, wird jede Textur  $\mathbf{G}$  in das Mittelwertshape  $\bar{\mathbf{x}}$  projiziert. Dazu wird ein Verfahren genutzt, welches mit Thin Plate Splines arbeitet, die 1989 von Bookstein ([2]) eingeführt wurden. Es wird im Anhang B näher beschrieben. Textur des so erzeugten Objektes sei der Vektor  $\mathbf{g}_{im}$ . Wirft man einen Blick auf verschiedene MRT-Hirnbilder, so ist die globale Helligkeit deutlich erkennbar nicht gleich. Um durch diesen störenden Effekt zu minimieren, werden die Texturen, wie von Cootes u. a. in [8] empfohlen, nun noch normalisiert:

$$\mathbf{g} = (\mathbf{g}_{im} - \beta \mathbf{1}) / \alpha. \quad (4.1)$$

Die Bestimmung von  $\alpha$  und  $\beta$  erfolgte zunächst nach Cootes u. a. ([8], S. 30) wie folgt: Ziel dabei ist es,  $\alpha$  und  $\beta$  so zu wählen, daß  $\mathbf{g}$  dem normalisierten Mittelwert  $\bar{\mathbf{g}}$  möglichst nahe ist. Die Summe von  $\bar{\mathbf{g}}$  soll 0 und die Varianz 1 sein. Dann ergeben

sich die Parameter zu

$$\alpha = \mathbf{g}_{im} \cdot \bar{\mathbf{g}} \quad , \quad \beta = (\mathbf{g}_{im} \cdot \mathbf{1}) / n, \quad (4.2)$$

wobei  $n$  die Anzahl der Pixel und  $\mathbf{1}$  ein Vektor von Einsen ist. Die praktische Bestimmung der Parameter erfolgt iterativ. Eine Textur wird als erste Schätzung für  $\bar{\mathbf{g}}$  gewählt<sup>1</sup>. Danach bestimmt man für jede Textur  $\alpha$  und  $\beta$  gemäß (4.2) und damit die jeweiligen normalisierten Texturen. Aus diesen wird der Mittelwert gebildet, der, erneut normalisiert<sup>2</sup>, die neue Schätzung für  $\bar{\mathbf{g}}$  bildet. Man iteriert, bis keine nennenswerte Änderung mehr stattfindet. Bei den vorliegenden Daten genügten dafür zwei bis drei Schritte.

Um die Suchergebnisse zu verbessern, wurde noch eine andere Variante erprobt, welche im Endeffekt etwas genauere Ergebnisse (S. 34) liefert. Bei dieser wird allerdings auf eine Anpassung der Helligkeitswerte während der Suche (29) verzichtet. Damit ist es zwar unmöglich, das Modell photorealistisch an das zugrundeliegende Suchbild anzupassen, letztlich genügt dieser Arbeit aber die Bestimmung der Form; die Textur ist nur Mittel zum Zweck.

Jede Textur wird einzeln normalisiert. Auf die Annäherung an den normalisierten Mittelwert der Texturen wird verzichtet. Als Skalierungsfaktor  $\alpha$  und Offset  $\beta$  werden die Standardabweichung und der Mittelwert der jeweiligen Textur verwendet.

Auf die normalisierten Vektoren  $\mathbf{g}_i$  wendet man nun eine PCA, wie von L. I. Smith in [21] beschrieben, an und erreicht damit ein Modell der Form

$$\mathbf{g} \approx \bar{\mathbf{g}} + \Phi_g \mathbf{b}_g, \quad (4.3)$$

wobei  $\bar{\mathbf{g}}$  der Mittelwert der normalisierten Texturen ist,  $\Phi_g$  die Varianz der Texturen durch die Eigenvektoren zu den  $t$  größten Eigenwerten der Kovarianzmatrix beschreibt und in  $\mathbf{b}_g$  die wählbaren Parameter des Texturmodells stehen.

---

<sup>1</sup>Von jedem Element dieser Textur wird der Mittelwert abgezogen, wonach es durch die Varianz dividiert wird

<sup>2</sup>Der Mittelwert muß immer 0 und die Varianz 1 sein

## 4.2 Das kombinierte Appearance-Modell

Mit den beiden Modellen

$$\mathbf{x} \approx \bar{\mathbf{x}} + \Phi_s \mathbf{b}_s \quad \text{und} \quad \mathbf{g} \approx \bar{\mathbf{g}} + \Phi_g \mathbf{b}_g \quad (4.4)$$

kann man nun neue komplette Objekte ähnlich den gegebenen erzeugen. Der besseren Handhabbarkeit wegen und um Korrelationen zwischen Shape- und Texturparametern zu eliminieren, wird nun aus diesen ein kombiniertes Modell erzeugt und einer erneuten PCA unterworfen [22]. Jedes Objekt des Trainingsdatensatzes wird dazu modelliert:

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s \Phi_s^T (\mathbf{x} - \bar{\mathbf{x}}) \\ \Phi_g^T (\mathbf{g} - \bar{\mathbf{g}}) \end{pmatrix}. \quad (4.5)$$

Die PCA der Vektoren  $\mathbf{b}$  ergibt das kombinierte Modell

$$\mathbf{b} \approx \Phi_c \mathbf{c}. \quad (4.6)$$

Da das Modell linearen Charakter hat, lassen sich Shape und Textur nun anhand *einer* Parameterliste  $\mathbf{c}$  daraus ermitteln:

$$\mathbf{x} = \bar{\mathbf{x}} + \Phi_s \mathbf{W}_s^{-1} \Phi_{cs} \mathbf{c} \quad (4.7)$$

$$\mathbf{g} = \bar{\mathbf{g}} + \Phi_g \Phi_{cg} \mathbf{c}, \quad (4.8)$$

mit

$$\Phi_c = (\Phi_{cs} \Phi_{cg})^T, \quad (4.9)$$

bzw.

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Q}_s \mathbf{c} \quad \text{mit} \quad \mathbf{Q}_s = \Phi_s \mathbf{W}_s^{-1} \Phi_{cs} \quad (4.10)$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c} \quad \text{mit} \quad \mathbf{Q}_g = \Phi_g \Phi_{cg}. \quad (4.11)$$

Die Gewichtsmatrix  $\mathbf{W}_s$  wichtet zwischen Pixel- bzw. Voxelabständen und Pixel- bzw. Voxelintensitäten. Der Einfachheit halber und weil ihr Einfluß gering ist, wurde die Gewichtsmatrix nach  $\mathbf{W}_s = r \mathbf{I}$  berechnet, wobei  $\mathbf{I}$  die Einheitsmatrix und  $r$  der Quotient aus Gesamtintensitätsvarianz und Gesamtshapevarianz der normalisierten Trainingsobjekte sind ([8], S. 32).

## 5 Die Objektsuche

Um ein Objekt der modellierten Klasse in einem Bild zu finden, ist es nötig, diejenigen Parameter zu ermitteln, welche das Modell so deformieren, daß es am besten mit jenem übereinstimmt. Um festzustellen, wie gut eine solcherart parametrisierte Modellinstanz zu dem Bild paßt, ist eine *Kostenfunktion*  $F$  zu definieren, die mißt, wie teuer eine Parameterkonstellation ist. Je größer der Abstand vom Optimum ist, desto teurer die Instanz. Das Ziel einer Suche ist es nun, diese Funktion zu optimieren. Für Appearancemodelle werden Fehlerfunktionen gewählt, die es zu minimieren gilt.

### 5.1 Suche in Shape-Modellen

Der Vollständigkeit wegen sei an dieser Stelle kurz auf die reine Shapesuche eingegangen. Für die Fehlerfunktion gibt es mehrere alternative Varianten. Man kann den durchschnittlichen Abstand zur nächsten starken Kante (deutlicher Helligkeitswechsel auf sehr kurzer Strecke) oder zu bekannten deutlich erkennbaren Strukturen messen. Da aber nicht immer die stärksten Kanten die Objektgrenze beschreiben müssen, ist es eine weitere Möglichkeit, sich während des Trainings die Grauwertverteilung um jeden Modellpunkt herum zu merken und während der Suche damit zu vergleichen.

Dazu beschreiben Cootes u. a. in [4] eine Variante, in der entlang des Normalenvektors zur bisher modellierten Grenze nach starken Intensitätswechsel gesucht wird und in [7] eine Möglichkeit, die Grauwertverteilung in einem Quadrat um jeden Punkt wieder als statistisches Modell zu trainieren. Während der Suche wird dann für jedes Pixel in der Umgebung eines (vorläufigen) Modellpunktes dessen Eignung aufgrund seiner Grauwertumgebung untersucht. Die Modellparameter  $\mathbf{b}_s$  müssen dann an die neuen Punkte angepaßt werden.

## 5.2 Suche in Appearance-Modellen

Im Gegensatz zu einem reinen Shape-Modell existieren hier mehr Informationen über das Objekt. Zu der Forminformation kommt auch noch die Texturinformation, die bei der Suche hilfreich sein wird.

Ziel des Suchalgorithmus' ist es nun nicht mehr nur die Form, sondern auch die Textur zu optimieren. Als Unterschied zwischen dem der Suche zugrundeliegenden Bild und der mit Parametern  $\mathbf{b}$  erzeugten Modellinstanz wird der Vektor  $\delta\mathbf{g}$  nach

$$\delta\mathbf{g} = \mathbf{g}_{Bild} - \mathbf{g}_{Modell} \quad (5.1)$$

definiert, wobei  $\mathbf{g}_{Bild}$  der Grauwertvektor des Bildes und  $\mathbf{g}_{Modell}$  der des Modells sind.  $\mathbf{g}_{Bild}$  wird natürlich nur in dem vom Shape  $\mathbf{x}_{Modell}$  umschlossenen Gebiet ermittelt, da die Grauwertvektoren gleich lang sein müssen.

Während des Suchprozesses ist es nun das Ziel, den totalen Wert der Grauwertdifferenz  $\Delta = |\delta\mathbf{g}|^2$  zu minimieren. Idealziel ist eine Differenz von 0, bei der man das Appearance-Modell perfekt auf das unterliegende Modell angepaßt hätte. Betrachtet man diese Modelle, so stellt man fest, daß die Anzahl der Parameter durchaus sehr groß sein kann, wodurch die Suche zunächst als ein hochdimensionales Optimierungsproblem erscheint. Cootes u. a. reduzieren dieses in [5] mit der Annahme, daß die Optimierungsversuche einander ähnlich sind. Bereits aus den Trainingsbildern wird daher ermittelt, wie man die Modellparameter generell anzupassen hat.

Da als Grundlage für die Optimierung das Muster der Grauwertverteilung in  $\delta\mathbf{g}$  gegeben ist, muß ein Zusammenhang zwischen  $\delta\mathbf{g}$  und der Änderung der Modellparameter gefunden werden.

Zur Erinnerung, das Appearance-Modell besteht aus zwei Gleichungen, je einer für Shape und Textur:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Q}_s \mathbf{c}, \quad (5.2)$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c}, \quad (5.3)$$

mit  $\bar{\mathbf{x}}$  als Mittelwertshape,  $\bar{\mathbf{g}}$  als Mittelwerttextur in  $\bar{\mathbf{x}}$ , dazu sind  $\mathbf{Q}_s$  und  $\mathbf{Q}_g$  Beschreibungen der Variation der Trainingsdaten aus dem kombinierten Appearance-Modell. Ein Shape  $\mathbf{x}$  ist noch normiert und hat seinen Schwerpunkt im Koordinatenursprung, muß also noch durch eine Transformation  $\mathbf{T}_s(\mathbf{x}, \mathbf{t})$ , wobei  $\mathbf{t}$  Trans-



formationsparameter sind, im Bild platziert werden. Dabei ist  $\mathbf{t}_{2D} = (a_x, a_y, t_x, t_y)$  mit  $a_x = s \cos(\theta)$ ,  $a_y = s \sin(\theta)$  und  $t_x, t_y$  als Verschiebungswerte ( $s$  ist dabei Skalierungsfaktor und  $\theta$  der Rotationswinkel). Dreidimensional ist, da auf Rotationsparameter verzichtet wurde,  $\mathbf{t}_{3D} = (s, t_x, t_y, t_z)$  mit  $s$  als Skalierungsfaktor,  $t_x, t_y$  und  $t_z$  sind Verschiebungswerte. Ein so entstandenes Shape sei im Folgenden als  $\mathbf{X}$  bezeichnet. Die Textur  $\mathbf{g}$  ist normalisiert. Eine Funktion  $\mathbf{T}_g(\mathbf{g}_{Bild}, \alpha_{g_{Bild}}, \beta_{g_{Bild}})$ , definiert wie (4.1), normalisiert die in  $\mathbf{X}$  eingelesene Textur  $\mathbf{g}_{Bild}$  des Bildes. Umgekehrt wandelt die Funktion

$$\mathbf{T}_g^{-1}(\mathbf{g}, \alpha, \beta) = \alpha \mathbf{g} + \beta \mathbf{1} \quad (5.4)$$

eine normalisierte Textur wieder in das Bildformat ( $\mathbf{1}$  ist ein Vektor aus Einsen). Solche Texturen sind im Folgenden mit  $\mathbf{G}$  bezeichnet, die Parameter  $\alpha$  (Texturskalierung) und  $\beta$  (Texturoffset) bilden den Texturparametervektor  $\mathbf{u} = (\alpha, \beta)^T$ .

### 5.2.1 Das Erlernen der Parametervorhersage

Aus dem Trainingsdatensatz kann nun, nachdem das kombinierte Appearance-Modell vorhanden ist, der gesuchte Zusammenhang zwischen  $\delta \mathbf{g}$  und der Änderung der Modellparametern  $\delta \mathbf{p}$  mit  $\mathbf{p}^T = (\mathbf{c}^T | \mathbf{t}^T | \mathbf{u}^T)$  errechnet werden. Der Vergleich zweier Texturen findet immer normalisiert statt, um globale Helligkeitsunterschiede zu minimieren, weiterhin werden Texturen dazu stets in das gleiche Shape, also nach  $\bar{\mathbf{x}}$  projiziert, Genaueres dazu steht in Anhang B.

Sei  $\mathbf{X}$  ein Shape, entstanden durch Parametern  $\mathbf{c}$  nach dem Modell (4.10) und der Transformation mit den Parametern  $\mathbf{t}$  nach (3.4) bzw. (3.7). Die Textur des Bildes in  $\mathbf{X}$  sei  $\mathbf{G}_{Bild}$ , nach der Projektion in  $\bar{\mathbf{x}}$  und Normalisierung ergibt sich  $\mathbf{g}_{Bild}$ . Die Modelltextur nach  $\mathbf{c}$  ergibt sich durch (4.11) zu  $\mathbf{g}_{Modell} = \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c}$ .

Sei nun die Differenz zwischen diesen Texturen

$$\mathbf{d}(\mathbf{p}) = \mathbf{g}_{Bild}(\mathbf{p}) - \mathbf{g}_{Modell}(\mathbf{c}). \quad (5.5)$$

Die Funktion  $\mathbf{d}$  zu optimieren bedeutet, eine Parameteränderung  $\delta \mathbf{p}$  zu finden, für die  $\mathbf{d}(\mathbf{p} + \delta \mathbf{p})$  minimal wird. Eine für diese Aufgabe durchgeführte Taylorentwicklung erster Ordnung um  $\mathbf{p}$  ergibt

$$\mathbf{d}(\mathbf{p} + \delta \mathbf{p}) \approx \mathbf{d}(\mathbf{p}) + \frac{\partial \mathbf{d}}{\partial \mathbf{p}} \delta \mathbf{p}. \quad (5.6)$$

Da das Optimum darin liegt,  $\delta \mathbf{p}$  so zu finden, daß  $\mathbf{d}(\mathbf{p} + \delta \mathbf{p}) = \vec{\mathbf{0}}$  wird, ergibt sich nach wenigen Umformungsschritten

$$\delta \mathbf{p} = -\mathbf{d}(\mathbf{p}) \cdot \mathbf{D} \quad \text{mit} \quad \mathbf{D} = \left( \left( \frac{\partial \mathbf{d}}{\partial \mathbf{p}} \right)^T \cdot \frac{\partial \mathbf{d}}{\partial \mathbf{p}} \right)^{-1} \cdot \frac{\partial \mathbf{d}}{\partial \mathbf{p}}. \quad (5.7)$$

Normalerweise müßte man  $\mathbf{D}$  für jede Iteration neu berechnen — eine sehr kostspielige Prozedur. Da aber  $\mathbf{D}$  normalisiert berechnet wird, also immer in dem selben Shape  $\bar{\mathbf{x}}$ , und mit normalisierten Texturen, wird gemäß Cootes und Taylor in [7] angenommen, daß  $\mathbf{D}$  ungefähr gleich bleibt. Das eröffnet die Möglichkeit, es, bzw.  $\frac{\partial \mathbf{d}}{\partial \mathbf{p}}$  einmalig aus den Trainingsdaten zu gewinnen und dann für jede Suche zu nutzen.

Die Berechnung von  $\frac{\partial \mathbf{d}}{\partial \mathbf{p}}$  erfolgt mittels numerischer Differentiation: Für jedes Trainingsbild wird von den bekannten optimalen Parametern ausgegangen. Diese können ohne weiteres durch Lösen eines linearen Gleichungssystems aus dem Modell und den bekannten Positionen der handgesetzten Landmarken bestimmt werden. Jeder der Parameter aus  $\mathbf{p}$  wird einzeln um verschiedene feste Beträge  $\delta p_{jk}$  ausgelenkt (die Indices stehen für Parameter und Auslenkung). Für jeden solchen Parametersatz  $(\mathbf{p} \pm \delta p_{jk})$  wird  $\mathbf{g}_{Modell}(\mathbf{p} \pm \delta p_{jk})$  berechnet (4.11). Weiter wird das zugehörige Shape  $\mathbf{x}_{Modell}(\mathbf{p} \pm \delta p_{jk})$  ermittelt (4.10), die Textur des Trainingsbildes innerhalb von  $\mathbf{x}_{Modell}(\mathbf{p} \pm \delta p_{jk})$  eingescannt und in das Mittelwertshape  $\bar{\mathbf{x}}$  projiziert. Das Ergebnis ist  $\mathbf{g}_{Bild}(\mathbf{p} \pm \delta p_{jk})$ . Nun wird der jeweilige Fehler  $\mathbf{d}(\mathbf{p} \pm \delta p_{jk})$  gemäß (5.5) bestimmt.

Ein Element der gesuchten Matrix  $\mathbf{D}$ ,  $\left( \frac{\partial \mathbf{d}}{\partial \mathbf{p}} \right)_{ij}$ , ist nun

$$\left( \frac{\partial \mathbf{d}}{\partial \mathbf{p}} \right)_{ij} = \sum_k \omega(\delta p_{jk}) \cdot (d_i(\mathbf{p} \pm \delta p_{jk}) - d_i(\mathbf{p})), \quad (5.8)$$

wobei  $i$  die Position innerhalb der Textur und  $j$  den ausgelenkten Parameter angeben. Bei  $\omega(\cdot)$  handelt es sich um einen normalisierenden Gaussfilter mit  $\omega(a) = \exp(a^2/2\sigma_a^2)/a$ , wobei  $\sigma_a^2$  die während des Trainingsprozesses ermittelte Varianz des Parameters angibt [6].

Es wurden verschieden starke Parameterauslenkungen getestet. Bei einem Modell eines Ventrikel (3.1, S. 14) erwiesen sich, wie von Cootes und Taylor ([8], S. 46) vorgeschlagen, Werte bis 0.5 der jeweiligen Standardabweichung für Appearance, Rotations- und Skalierungsparameter als beste Wahl. Lediglich die Auslenkung

der Translationsparameter ergab für  $\pm 5$  Pixel (anstatt 3) bessere Ergebnisse und für den Texturskalierungsparameter ergaben die empfohlenen Werte von  $\pm 10\%$  sehr schlechte Ergebnisse, Auslenkungen bis max. 1% waren günstiger. Dies liegt an der deutlich geringeren Gesamttexturvariation von einzelnen Hirnobjekten im Vergleich zu Gesichtern.

Bei Versuchen mit einem Modell des Putamen erwiesen sich hingegen andere Auslenkungen als wesentlich besser. Hier wurden die Appearanceparameter bis zu einer Standardabweichung ausgelenkt, ebenso Rotations- und Skalierungsparameter. Die Translationsparameter wurden bis zu 3 Pixel variiert.

Die Unterschiede zwischen den beiden Modellen liegen vor allem darin, daß Ventrikel auf den vorliegenden Bildern einen sehr deutlichen Unterschied zum umliegenden Gewebe haben (weshalb hier kantenbasierte Verfahren sicher bessere Ergebnisse erbringen würden). Hingegen befindet sich das Putamen in einer deutlich weniger abgesetzten Umgebung. Weiterhin ist die Shapevariation im Ventrikelmodell deutlich stärker ausgeprägt.

Es ist für jedes neue Modell zu überprüfen, ob die gewählten Parameter optimierbar sind. Variationen dieser Auslenkungen erbringen z. T. große Unterschiede in den Ergebnisqualität.

### 5.2.2 Modellanpassung

Mit (5.7) ist ein Mittel gegeben, um das Modell an Bilder anzupassen. Dies erfolgt mittels eines iterativen Algorithmus' (nach [6]). Im ersten Schritt legt man durch  $\mathbf{t}_0$  die mittlere Position der Trainingsshapes fest, um eine sinnvolle Anfangsposition zu erhalten. Die Appearance-Parameter  $c_i$  sind alle gleich 0. Eine Iteration im Anpassungsalgorithmus erfolgt so (die aktuellen Parameter sind  $\mathbf{p}_0$ ):

1. Berechne Modellshape  $\mathbf{X}_{Modell_0}$  und -textur  $\mathbf{g}_{Modell_0}$  nach (4.10) bzw. (4.11).
2. Ermittle Bildtextur  $\mathbf{G}_{Bild_0}$  in dem von  $\mathbf{X}_{Modell_0}$  umschlossenen Bereich und projiziere sie nach  $\bar{\mathbf{x}}$ , die neue Textur wird noch normalisiert und ist dann  $\mathbf{g}_{Bild_0}$ .
3. Berechne den Differenzvektor  $d(\mathbf{p}_0) = \mathbf{g}_{Bild_0} - \mathbf{g}_{Modell_0}$  und den Fehlerwert  $E_0 = |\delta \mathbf{d}(\mathbf{p}_0)|^2$ .

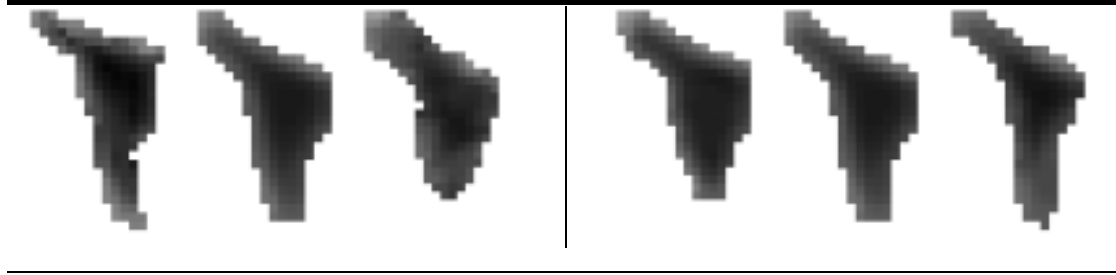


Abbildung 5.1: Mittelwertmodell für eine Ventrikelschicht: Auslenkung des 1. bzw. 2. Parameters um je  $\pm 2$  Standardabweichungen

---

4. Bestimme die Parameteränderung nach (5.7) mit  $\delta \mathbf{p} = -d(\mathbf{p}) \cdot \mathbf{D}$
5. Setze  $k = 1$
6. Die neue Parameterschätzung lautet  $\mathbf{p}_1 = \mathbf{p}_0 + k\delta \mathbf{p}$
7. Berechne mit  $\mathbf{p}_1$  eine neue Modellinstanz  $\mathbf{X}_{Modell_1}$  und  $\mathbf{g}_{Modell_1}$
8. Scanne eine neue Bildtextur in  $\mathbf{X}_{Modell_1}$ , projiziere diese nach  $\bar{\mathbf{x}}$  und normalisiere sie; das Ergebnis ist  $\mathbf{g}_{Bild_1}$
9. Bestimme den neuen Fehlervektor  $d(\mathbf{p}_1) = \mathbf{g}_{Bild_1} - \mathbf{g}_{Modell_1}$  und den neuen Fehlerwert  $E_1 = |\delta \mathbf{d}(\mathbf{p}_1)|^2$ .
10. Wenn  $E_1 < E_0$  ist, dann ist die neue Schätzung besser als die alte und es wird mit ihr weiter verfahren: Setze  $\mathbf{p}_0 := \mathbf{p}_1$  und beginne erneut bei 1.
11. Wenn nicht, dann setze  $k = k/2$  und gehe zu 6.

Diese Iteration wird so oft vollzogen, bis keine signifikante Verbesserung mehr erreicht wird, also bis  $k$  einen genügend kleinen Wert (z. B.  $1/32$ ) erreicht hat.

### 5.2.3 Modelleinschätzung

Um zu erforschen, in welchen Parameterbereichen die Suchfunktion bei einem Modell funktioniert, testet man wieder an den Trainingsbildern. Von diesen sind die optimalen Parameter bekannt. Gegeben sind also für jedes Experiment die Modellparameter und das zugrundeliegende Bild. Man lenkt man nun einen der Parameter

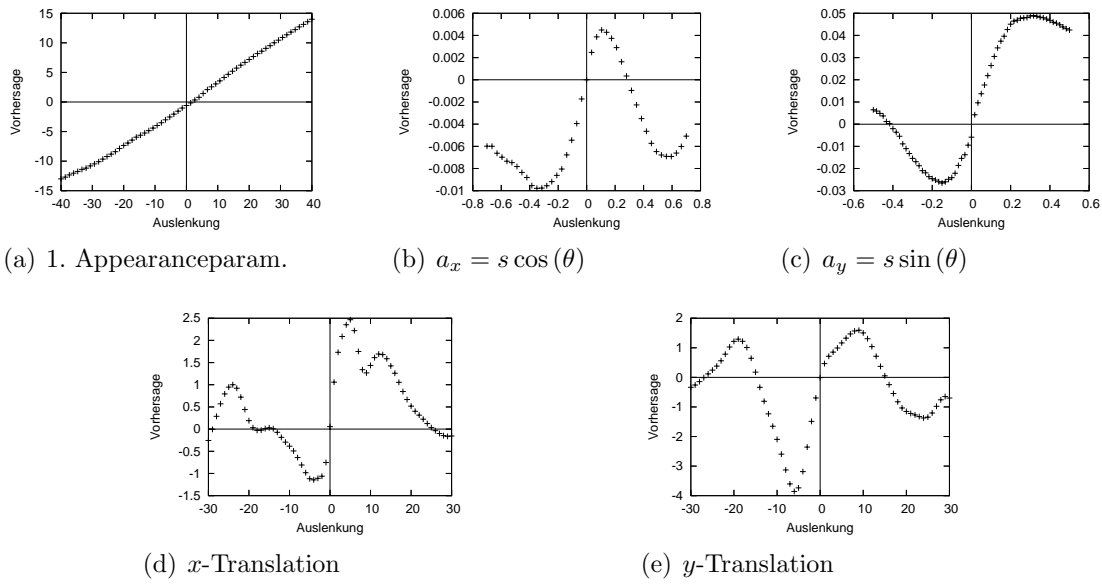


Abbildung 5.2: Darstellung von Auslenkung und der Vorhersage der 1. Iteration für den stärksten Appearance-Parameter und vier lineare Transformationsparameter:  $a_x = s \cos(\theta)$ ,  $a_y = s \sin(\theta)$ ,  $x$ - und  $y$ -Translation. Alle Darstellungen betreffen ein Ventrikelmodell und das Mittel über 10 Testbilder

um einen bestimmten Wert aus. Mit dem neuen Parametersatz erzeugt man eine Modellinstanz und schreibt sie in das Bild hinein. Das so gewonnene neue Bild bearbeitet man mit dem auf den in Frage kommenden Parameter eingeschränkten Suchalgorithmus. Dies vollzieht man mit verschiedenen Parameterauslenkungen. Die vom Modell vorhergesagten Änderungen des Parameters kann man nun mit den tatsächlichen in ein Diagramm eintragen und daraus Folgerungen ziehen. Abb. 5.2 zeigt einige solcher Diagramme. In den Bereichen, in denen Auslenkung und Vorhersage das gleiche Vorzeichen haben, findet eine Annäherung an die tatsächliche Parameterkonfiguration statt. Außerhalb dieser Bereiche entfernt sich das Modell vom Ideal und findet andere, lokale Optima, die aber nicht erwünscht sind. So zeigen beispielsweise 5.2(d) und 5.2(e), daß Objekte, die weiter als ca. 20 Pixel vom Erwartungsort entfernt sind, nicht mehr angenähert werden können. Ist dieser Wert zu klein, so sollte bei der Bestimmung der Parametervorhersagematrix optimiert werden.

## 6 Implementation

Zur praktischen Anwendung der beschriebenen Algorithmen wurden Programme in der Sprache C geschrieben. Jedes existiert in einer Version zwei- und in einer für dreidimensionale Berechnungen. Für die Programme zur Behandlung zweidimensionaler Modelle gibt es eine Unterscheidung zwischen zwei verschiedenen Normalisierungsmöglichkeiten (4.1).

Das Trainingsprogramm benötigt als Ausgangsdaten die Trainingsbilder<sup>1</sup> und Dateien mit den zugehörigen, von Hand ermittelten, Landmarkenpunkten. Anhand dieser erstellt es zunächst ein Shape-Modell (3.14), danach aus den Texturen ein Texturmodell (4.3) und vereint dann beide Modelle zu einem Appearance-Modell (4.10, 4.11). Am Ende wird noch die Parametervorhersagematrix  $\mathbf{D}$  (5.7) ermittelt. Als Ausgabe werden das mittlere Shape, die mittlere Textur und die Basismatrizen  $Q_s$  und  $Q_g$  für das Appearance-Modell geliefert. Weiterhin werden Dateien mit den Modellparametern der Trainingsbilder abgelegt, die u. U. bei der Suchfunktion benötigt werden.

Mit diesen Daten kann man durch ein Testprogramm, wie in 5.2.3 beschrieben, bestimmen, in welchem ungefähren Rahmen die Modellsuche Verbesserungen zeitigen wird. Dazu wird ein Trainingsbild gewählt, für welches die Modellparameter bekannt sind. Nun lassen sich einzelne Parameter<sup>2</sup> auslenken und man beobachtet, inwieweit die auf diesen Parameter eingeschränkte Modelloptimierungsfunktion wirkt. Durch Anwendung verschiedener Auslenkungen auf eine Menge von Bildern ergibt sich ein Bild der Suchmöglichkeiten.

Das Suchprogramm schließlich ermittelt aus den Ergebnissen des Trainings und einem neuen Bild iterativ eine Schätzung von Modellparametern. Diese führen zur Bestimmung einer neuen Modellinstanz — eine Menge von Landmarken, das Ergebnis der Suche. Es gibt zwei Möglichkeiten der Suche. Die erste Variante startet mit

---

<sup>1</sup>Für den zweidimensionalen Fall existiert ein Programm zur Extraktion einer beliebigen Schicht aus einem 3D-Bild

<sup>2</sup>Natürlich kann man auch mehrere Parameter gleichzeitig testen

dem Mittelwertmodell als Grundlage. Die zweite prüft erst eine Menge von Templates, vordefinierte Modellinstanzen mit bekannter Parameterliste<sup>3</sup>. Das Muster mit dem geringsten quadratischen Grauwertfehler zwischen seiner Textur und dem Bild wird als Grundlage für die Suche genutzt. Dabei ist es nicht nötig, das Muster in einem gewissen Rahmen über das Bild zu schieben und jeweils die Texturunterschiede zu berechnen. Stattdessen wird der AAM-Suchalgorithmus, eingeschränkt auf die Verschiebungsparameter angewandt. Dies minimiert die Suchzeit extrem — pro Muster waren wenige Iterationen nötig, da der auf diese Parameter eingeschränkte Suchalgorithmus sehr schnell und genau konvergiert.

---

<sup>3</sup>Diese stammen aus dem Trainingsprozeß oder vorangegangenen erfolgreichen Suchprozessen.

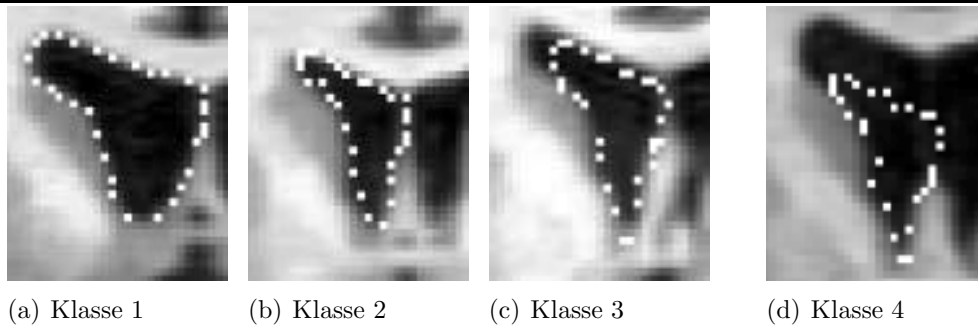


Abbildung 7.1: Ein Beispiel für die Klasseneinteilung der Suchergebnisse

---

## 7 Ergebnisse

Bei Versuchen wurden die ermittelten Ergebnisse nach visueller Begutachtung in 4 Klassen eingeteilt.

1. sehr gute Ergebnisse (vgl. Abb. 7.1(a))
2. Ergebnisse mit leichten Mängeln, wenn z. B. eine Kante knapp verfehlt wurde (vgl. Abb. 7.1(b))
3. Ergebnisse mit stärkeren Mängeln (vgl. Abb. 7.1(c))
4. Ergebnisse, bei denen das Objekt sehr schlecht wiedergegeben wurde (vgl. Abb. 7.1(d))

Um zu ermitteln, wie das Verfahren mit Objekten, die sehr klar zu erkennen sind, zurechtkommt, wurde ein zunächst Modell des Schnittes durch ein Ventrikel betrachtet (Abb. 7.1). Dieses zeichnet sich durch einen recht starken Unterschied zum umliegenden Gewebe aus. Die Formen dieses Objektes sind recht variabel ausgeprägt. Es gibt starke Variation in der Breite und in der Länge vor allem der oberen linken Ausbuchtung. Erstellt wurden mehrere Modelle, in die verschieden viele Trainingsbilder



einfließen, um zu untersuchen, wie sich eine Änderung der Modellgröße auf die Ergebnisgüte auswirkt. Es wurde mit 10, 35 und 70 handsegmentierten Trainingsbildern gearbeitet. Betrachtet wurden 200 Bilder, die nicht im Trainingsset enthalten waren. Es wurden beide Varianten der Normalisierung angewandt und jeweils zwischen Durchläufen mit direkter Anpassung und solchen mit vorgeschobener Mustersuche unterschieden. Beim Training wurden bei der Berechnung der Parametervorhersagematrix (5.2.1, S. 27) folgende Parameteraulenkungen verwendet. Modellparameter: bis 0.5· Standardabweichung, Rotations- und Skalierungsparameter: bis 0.5· Standardabweichung, Translationsparameter: bis 5 Pixel, Texturskalierung: bis 1%. Versuche mit anderen Werten ergaben keine Verbesserungen.

Für die von Cootes u. a. [8] vorgeschlagene Normalisierung (im weiteren mit A bezeichnet) ergab sich:

Klasse	1	2	3	4
Anzahl ohne Vorstufe (mit 10 Trainingsbildern)	23	102	54	21
Anzahl mit Vorstufe (mit 10 Trainingsbildern)	32	96	58	14
Anzahl ohne Vorstufe (mit 35 Trainingsbildern)	25	83	70	22
Anzahl mit Vorstufe (mit 35 Trainingsbildern)	35	103	55	7
Anzahl ohne Vorstufe (mit 70 Trainingsbildern)	37	74	59	30
Anzahl mit Vorstufe (mit 70 Trainingsbildern)	38	93	62	7

Suchläufe mit der experimentell optimierten Normalisierung (4.1, S. 22, im weiteren mit B bezeichnet) lieferten:

Klasse	1	2	3	4
Anzahl ohne Vorstufe (mit 10 Trainingsbildern)	32	94	60	14
Anzahl mit Vorstufe (mit 10 Trainingsbildern)	33	90	70	7
Anzahl ohne Vorstufe (mit 35 Trainingsbildern)	41	94	48	20
Anzahl mit Vorstufe (mit 35 Trainingsbildern)	35	111	49	5
Anzahl ohne Vorstufe (mit 70 Trainingsbildern)	49	82	45	24
Anzahl mit Vorstufe (mit 70 Trainingsbildern)	67	80	41	12

Es ist zu bemerken, daß die verschiedenen Varianten nur in ca. 50% der Fälle einem Bild die gleiche Klasse zuordnen. Gerade bei Zuordnungen der mittleren Klassen treten oft Unterschiede auf. Die größten Probleme traten bei der Erkennung der oberen linken Ausbuchtung auf.

Deutlich am schnellsten konvergiert die Variante mit Cootes' Normalisierung — es werden selten mehr als 3 Iterationen benötigt. Die Anpassung erfolgt innerhalb von Millisekunden (auf einem PC mit 2.6 GHz). Die zweite, genauigkeitsorientiert offenbar bessere Option konvergiert durchschnittlich in 13 Schritten. Dabei werden 1-2 Sekunden beansprucht. Anwendung mit vorheriger Mustersuche liefern durchschnittlich etwas bessere Ergebnisse, benötigen jedoch mehrere Sekunden für die Vorstufe. Dabei diese verläuft diese vor allem bei Normalisierung A sehr schnell — 70 Templates wurden innerhalb von ca. 3 Sekunden durchsucht.

Der quadratische Fehler zwischen Ergebnistextur und Bildtextur innerhalb der Modellform ist bei der Normalisierung A deutlich geringer. Er liegt bei guten bis sehr guten Ergebnissen zwischen 0.2 und 0.6. Dagegen liefert das Modell bei Normalisierung B für sehr gute Ergebnisse aufgrund des Verzichts auf Texturoptimierung Fehlerwerte zwischen 10 und 35. Diese Fehlerwerte sind allerdings nicht als sinnvolles Kriterium für den Sucherfolg zu werten. Gerade fehlerhafte Suchläufe, bei denen das Ergebnis im Inneren des Objektes liegt, liefern ebenfalls geringe Fehler, ebenso wie Fälle, in denen die Texturunterschiede zwischen benachbarten Arealen gering sind. Versuche, dieses Kriterium und die Sucherfolge i. A. zu optimieren, indem Modelle erzeugt wurden, bei denen die Landmarken ein wenig (ca. 1-2 Pixel) außerhalb des eigentlichen Randes gesetzt wurden, führen zu sehr schlechten Ergebnissen, meist mit einer starken Überschätzung des tatsächlichen Objekts einhergehend.

Wählt man aus den vier verschiedenen Suchergebnissen für ein Bild das jeweils beste aus, so ergibt sich eine Verteilung von:

Klasse	1	2	3	4
Anzahl bestes Ergebnis (mit 10 Trainingsbildern)	76	97	25	2
Anzahl bestes Ergebnis (mit 35 Trainingsbildern)	78	105	12	5
Anzahl bestes Ergebnis (mit 70 Trainingsbildern)	109	72	14	5

Leider wurde aber keine Möglichkeit gefunden, die Qualität anders als durch visuelle Begutachtung zu ermitteln.

Eine Untersuchung, bei wievielen Bildern erkennbare Verbesserungen oder Verschlechterungen (als Veränderungen ihrer Klasseneinstufung) durch Anwendung der Vorverarbeitung eintraten, lieferte folgende Ergebnisse:

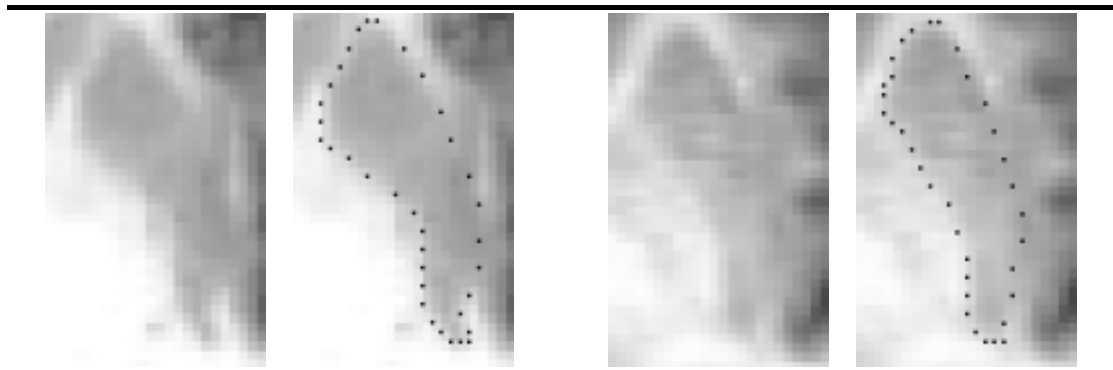


Abbildung 7.2: Eine Verdeutlichung der Problematik, Gewebegrenzen klar zu definieren, am Beispiel zweier Putamen

Qualitätsänderung	Verbesserung	unverändert	Verschlechterung
Normalisierung A (10 Bilder)	50	102	48
Normalisierung B (10 Bilder)	54	102	44
Normalisierung A (35 Bilder)	95	71	34
Normalisierung B (35 Bilder)	97	59	44
Normalisierung A (70 Bilder)	71	82	48
Normalisierung B (70 Bilder)	70	87	43

Typischerweise profitieren Bilder, die bei der reinen Suche sehr schlecht abschnitten, überdurchschnittlich stark von der Vorstufe, während vor allem sehr gut erkannte Bilder dabei nicht selten schlechtere Ergebnisse lieferten.

Als zweites Modell wurde ein Schnitt durch ein Putamen untersucht (Abb. 7.2). Dieses weist einen z. T. sehr geringen Grauwertunterschied zum umliegenden Gewebe aus, was es schon für einen Menschen nicht leicht macht, es korrekt abzugrenzen. Die Formvarianten sind weniger vielfältig, aber mit der unteren Spitze ist eine kleine Struktur gegeben, die in verschiedenen feiner Ausbildungen zu finden ist. An dieser Stelle traten die häufigsten Suchfehler auf. Die während des Trainings ermittelten Standardabweichungen waren für die stärksten Parameter im Shapemodell deutlich geringer als beim Ventrikelmodell, was eine geringere Formvarianz verdeutlicht

Für dieses Modell ergab sich bei der Suche ein sehr starker Unterschied bzgl. der Stärke der im Training angewandten Parameterauslenkungen (5.2.1, S. 27). Für die Standardwerte ([8], S. 46), die sich auch beim Ventrikelmodell als optimal erwiesen (Appearance-, Rotations- und Skalierungsparameter: bis 0.5· Standardabweichung,

Translationsparameter: bis 6 Pixel, Texturskalierung: bis 1%) ergaben Trefferquoten bis 25% für Klasse 1 und 50% für Klasse 2 (bei paralleler Anwendung aller Varianten bis 40%/45%). Darum wurde mit verschiedenen anderen Trainingsparametern getestet. Als optimale Wahl erwies sich eine Auslenkung der Appearance-, Rotations- und Skalierungsparameter bis 1.0· Standardabweichung, der Translationsparameter bis 3 Pixel und der Texturskalierung bis 1%). Für ein Testset aus 200 Bildern ergab sich:

Klasse	1	2	3	4
Anzahl ohne Vorstufe (Normalisierung A, 10 Trainingsbilder)	29	158	9	4
Anzahl mit Vorstufe (Normalisierung A, 10 Trainingsbilder)	51	128	16	5
Anzahl ohne Vorstufe (Normalisierung B, 10 Trainingsbilder)	85	97	15	3
Anzahl mit Vorstufe (Normalisierung B, 10 Trainingsbilder)	78	100	12	10
Anzahl ohne Vorstufe (Normalisierung A, 35 Trainingsbilder)	43	146	8	3
Anzahl mit Vorstufe (Normalisierung A, 35 Trainingsbilder)	48	135	15	2
Anzahl ohne Vorstufe (Normalisierung B, 35 Trainingsbilder)	98	93	6	3
Anzahl mit Vorstufe (Normalisierung B, 35 Trainingsbilder)	72	115	10	3
Anzahl ohne Vorstufe (Normalisierung A, 70 Trainingsbilder)	50	129	17	4
Anzahl mit Vorstufe (Normalisierung A, 70 Trainingsbilder)	33	140	25	2
Anzahl ohne Vorstufe (Normalisierung B, 70 Trainingsbilder)	121	63	10	6
Anzahl mit Vorstufe (Normalisierung B, 70 Trainingsbilder)	78	115	5	2

Zusammengefaßt ergeben sich die jeweils besten Ergebnisse aller vier Varianten zu:

Klasse	1	2	3	4
Anzahl bestes Ergebnis (mit 10 Trainingsbildern)	111	84	5	0
Anzahl bestes Ergebnis (mit 35 Trainingsbildern)	121	77	2	0
Anzahl bestes Ergebnis (mit 70 Trainingsbildern)	140	57	3	0

Die abfallenden Ergebnisse bei vorgeschaltetem Abgleich mit bekannten Bildern lassen sich dadurch erklären, daß der Grauwertunterschied zu benachbarten Arealen so gering ist, daß falsche Muster mitunter sehr geringe Fehlerwerte aufweisen. Dies geschieht z. B., wenn sich ein Muster zu weit rechts vom wirklichen Objekt befindet. Dort ist der Texturunterschied gering genug, daß eine korrekte Anpassung problematisch wird. Dieses Problem besteht beispielsweise beim Ventrikelmodell nicht, da

dort der Grauwertunterschied deutlich größer ausfällt, weshalb abseits liegende Muster große Fehlerwerte aufweisen.

Generell dauert die Suche bei diesem Modell etwas länger, da mehr Grauwerte zu verarbeiten sind. Außerdem wurde die Parameterauslenkung für die Translation geringer gewählt, wodurch vor allem die Vorstufe mehr Iterationen benötigt. So erfordert der Abgleich mit 70 Trainingsbildern ca. 20 Sekunden. Indes arbeitet die Variante ohne Vorstufe hier deutlich besser. Jene ist nur anzuraten, um schlechte Ergebnisse zu verbessern.

Modelle, die mehrere Objekte beinhalten, um auch Lagebeziehungen zwischen diesen zu beachten, lieferten keine guten Ergebnisse.

Die Möglichkeit, die Parameteranzahl der Modelle durch Anwendung der PCA (3.3, S. 20) zu minimieren funktioniert sehr gut. Die Beschränkung auf 90% der erklärten Varianz liefert noch sehr gute Ergebnisse, eine weitere Minierung auf 80% lief bei nur 4 Appearanceparametern nur wenig schlechter. In der Praxis bedeutet dies nur eine Minimierung der Trainingszeit. Die Suchgeschwindigkeit wird davon kaum forciert.

Ein kleines 3D-Modell des linken Ventrikels (Abb. 3.2, S. 15) wurde erzeugt, um die implementierte Erweiterung des 2D-Algorithmus' zu testen. Dieses Objekt ist eigentlich ungeeignet, da Teile in vielen Bildern nicht oder nur teilweise existierten (die beiden kleinen hervorstehenden Bereiche ganz rechts im Beispiel ebenso wie der linke Ausläufer) oder aufgrund zu geringer Auflösung (160x160x200) nicht erkennbar waren. Damit sind korrespondierende Punkte nicht ermittelbar. Gewählt wurde es dennoch, weil andere Objekte mangels genauer anatomischer Kenntnisse und Erfahrung nicht gut genug abzugrenzen waren. Darum wurde es auch lediglich zu Funktionalitätstests der Implementation genutzt.

Enthalten sind 239 Punkte, die Anzahl der Texturvoxel beläuft sich auf 10.000-22.000. Die manuelle Segmentation (schichtweise 2D-Markierung und anschließendes Positionieren der Objektlandmarken am 3D-Objekt) eines Objektes benötigte Stunden. Weiterhin benötigen einige Funktionen sehr viel Zeit. An erster Stelle steht hier das Textursampling, also die Erkennung, welche Voxel sich innerhalb der Sphäre befinden (Anhang C, S. 52). Diese benötigt je nach Objektgröße 1-4 Minuten und ist in jeder Iteration durchzuführen. Ein Trainingsvorgang dauerte (vor allem aufgrund der Bestimmung der Parametervorhersagematrix) über eine Stunde. Aufgrund des immensen Zeitaufwandes wurde von weiteren 3D-Modellen und intensiven Testrei-

hen Abstand genommen. Anwendungen der Suchfunktion (10-15 Minuten) auf neue Bilder blieben schon aufgrund der geringen Modellgröße erfolglos. Prinzipiell funktioniert der Algorithmus aber — Experimente mit Auslenkung einzelner Parameter vom bekannten Original (5.2.3) ergaben die richtigen Korrekturen durch die Suchfunktion

## 8 Diskussion

Active Appearance Models erwiesen sich bei den vorliegenden MRT-Gehirndaten (Schichten aus einem 160x160x200 - Bild) nur bedingt als einsetzbar. Für die Güte der Segmentation ist neben Anzahl und Qualität der Trainingsdaten vor allem die Formvariation der betreffenden Objektklasse verantwortlich. Am Beispiel von Ventrikelschnitten ergab sich trotz klarer Kanten und deutlichem Grauwertunterschied zu benachbarten Gebieten bei der günstigsten Trainings- und Suchvariante lediglich eine Quote von 34% sehr guten und 40% guten Suchergebnissen und auch dies nur bei vorherigem Abgleich mit den bekannten Trainingsbildern. Dagegen ergab das Beispiel von Schnitten durch das deutlich weniger formvariable Putamen Quoten von 60% sehr guten und 31% guten Ergebnissen ohne vorherigem Abgleich, obwohl die Grauwertdifferenz zu Nachbargewebe wesentlich geringer ausfällt. Dies ließ sich durch Betrachtung aller vier Suchvarianten pro Bild bis 55%/36% (Ventrikel) bzw. 70%/28% (Putamen) steigern, jedoch ohne die Möglichkeit, das jeweils beste der vier Ergebnisse automatisch zu detektieren.

Eine weitere Nutzungsmöglichkeit dieser Modellierungsalgorithmen fiel bei den Versuchen, die Ergebnisse zu optimieren auf. Betrachtet man einen gegebenen Modellobjekt mit zugehörigem Parametersatz und nur Anpassung von Translation, Skalierung und Rotation, so findet eine sehr schnelle Anpassung an das optimale Resultat statt. Dieses unterscheidet sich kaum vom naiven Vorgehen, das Objekt über ein Bild zu schieben und sich die beste Position zu merken. Auf diese Weise läßt sich eine Datenbank schnell auf das bestmögliche Muster durchsuchen, um von diesem Ausgangspunkt ggf. mit einer anderen Modellierungsmethode weiterzusuchen.

Zu dem Versuch einer dreidimensionalen AAM-Modellierung ist zu bemerken, daß für eine praktische Erprobung mit dieser Implementation zunächst die Programmeffizienz bei der Volumenbestimmung gesteigert werden muß. Für Objekte mit einigermaßen klaren Kanten sind Erweiterungen von Active Shape Models [14] oder Feder-Masse-Modelle [10] sicher eine effizientere Wahl.

# A Aligning zweier Shapes

## A.1 Aligning zweier 2D Shapes

Gegeben sei ein Paar ähnlicher Shapes  $\mathbf{x}$  und  $\mathbf{x}'$ , die beide ihren Schwerpunkt im Koordinatenursprung haben. Ziel ist es nun, Shape  $\mathbf{x}$  durch Rotation um  $\theta$ , Skalierung um  $s$  und Translation um  $(t_x, t_y)$  möglichst nah an Shape  $\mathbf{x}'$  anzunähern:

$$\mathbf{T}(s, \theta, t_x, t_y) \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} (s \cos \theta) x_i - (s \sin \theta) y_i + t_x \\ (s \sin \theta) x_i + (s \cos \theta) y_i + t_y \end{pmatrix}. \quad (\text{A.1})$$

Setze

$$a_x = (s \cos \theta) \quad a_y = (s \sin \theta). \quad (\text{A.2})$$

Zur Ermittlung der optimalen Parameter wird dazu der gewichtete quadratische Fehler minimiert:

$$\begin{aligned} E &= (\mathbf{T}(\mathbf{x}) - \mathbf{x}')^T \mathbf{W} (\mathbf{T}(\mathbf{x}) - \mathbf{x}') \\ &= \sum_{i=1}^n (a_x x_i - a_y y_i + t_x - x'_i)^2 w_i + (a_y x_i + a_x y_i + t_y - y'_i)^2 w_i. \end{aligned} \quad (\text{A.3})$$

Durch Differenzierung nach den gesuchten Parametern ergibt sich ein lineares Gleichungssystem:



$$\begin{aligned}
\frac{dE}{da_x} = 0 &= 2a_x \sum_{i=1}^n (x_i^2 + y_i^2) w_i + 2t_x \sum_{i=1}^n x_i w_i + 2t_y \sum_{i=1}^n y_i w_i - \\
&\quad 2 \sum_{i=1}^n (x_i x'_i + y_i y'_i) w_i \\
\frac{dE}{da_y} = 0 &= 2a_y \sum_{i=1}^n (x_i^2 + y_i^2) w_i - 2t_x \sum_{i=1}^n y_i w_i + 2t_y \sum_{i=1}^n x_i w_i + \\
&\quad 2 \sum_{i=1}^n (y_i x'_i - x_i y'_i) w_i \\
\frac{dE}{dt_x} = 0 &= 2a_x \sum_{i=1}^n x_i w_i - 2a_y \sum_{i=1}^n y_i w_i + 2t_x \sum_{i=1}^n w_i - 2 \sum_{i=1}^n x'_i w_i \\
\frac{dE}{dt_y} = 0 &= 2a_x \sum_{i=1}^n y_i w_i + 2a_y \sum_{i=1}^n x_i w_i + 2t_y \sum_{i=1}^n w_i - 2 \sum_{i=1}^n y'_i w_i.
\end{aligned} \tag{A.4}$$

Zur Vereinfachung seien:

$$\begin{aligned}
\mathbf{S}_x &= \sum_{i=1}^n x_i w_i & \mathbf{S}_{x'} &= \sum_{i=1}^n x'_i w_i \\
\mathbf{S}_y &= \sum_{i=1}^n y_i w_i & \mathbf{S}_{y'} &= \sum_{i=1}^n y'_i w_i \\
\mathbf{S}_{xx'+yy'} &= \sum_{i=1}^n (x_i x'_i + y_i y'_i) w_i & \mathbf{W} &= \sum_{i=1}^n w_i \\
\mathbf{S}_{xy'-yx'} &= \sum_{i=1}^n (x_i y'_i - y_i x'_i) w_i & \mathbf{S}_{xx+yy} &= \sum_{i=1}^n (x_i^2 + y_i^2) w_i
\end{aligned} \tag{A.5}$$

und damit

$$\begin{pmatrix} \mathbf{S}_x & -\mathbf{S}_y & \mathbf{W} & 0 \\ \mathbf{S}_y & \mathbf{S}_x & 0 & \mathbf{W} \\ \mathbf{S}_{xx+yy} & 0 & \mathbf{S}_x & \mathbf{S}_y \\ 0 & \mathbf{S}_{xx+yy} & -\mathbf{S}_y & \mathbf{S}_x \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \mathbf{S}_{x'} \\ \mathbf{S}_{y'} \\ \mathbf{S}_{xx'+yy'} \\ \mathbf{S}_{xy'-yx'} \end{pmatrix}, \tag{A.6}$$

woraus sich die Parameter  $a_x$ ,  $a_y$ ,  $t_x$  und  $t_y$  ohne weiteres berechnen lassen.

## A.2 Aligning zweier 3D Shapes

Das folgende Alignment-Verfahren ignoriert die Möglichkeit der Rotation. Möglich ist dies, weil die zur Verfügung stehenden MRT-Bilder bereits einer Prozedur unterzogen wurden, die sie in das Talairach-Koordinatensystem [9] überführt hat. Damit

sind gerade in der Nähe der Hirnmitte befindliche Objekte bereits so ähnlich orientiert, daß eine weitere Rotationsoptimierung nicht vonnöten ist. Da die Berechnungen für dreidimensionale Modelle sehr zeitintensiv sind, wurde an dieser Stelle gespart. Für andere Anwendungen ist es ein Leichtes, den für Interpolationszwecke implementierten Algorithmus von Arun (S. 49) an diese Stelle zu setzen.

Ohne Rotation zu erlauben, ergibt sich der quadratische Fehler wie folgt:

$$\begin{aligned} E &= (\mathbf{T}(\mathbf{x}) - \mathbf{x}')^T \mathbf{W} (\mathbf{T}(\mathbf{x}) - \mathbf{x}') \\ &= \sum_{i=1}^n (sx_i + t_x - x'_i)^2 w_i + (sx_i + t_y - y'_i)^2 w_i + (sz_i + t_z - z'_i)^2 w_i. \end{aligned}$$

Durch Differenzierung nach den gesuchten Parametern ergibt sich ein lineares Gleichungssystem, welches, 0 gesetzt, zu lösen ist:

$$\begin{aligned} \frac{dE}{ds} = 0 &= 2s \left( \sum_{i=1}^n x_i^2 w_i + \sum_{i=1}^n y_i^2 w_i + \sum_{i=1}^n z_i^2 w_i \right) + \\ &\quad 2t_x \sum_{i=1}^n x_i w_i + 2t_y \sum_{i=1}^n y_i w_i + 2t_z \sum_{i=1}^n z_i w_i - \\ &\quad 2 \sum_{i=1}^n x_i x'_i w_i - 2 \sum_{i=1}^n y_i y'_i w_i - 2 \sum_{i=1}^n z_i z'_i w_i \\ \frac{dE}{dt_x} = 0 &= 2s \sum_{i=1}^n x_i w_i + 2t_x \sum_{i=1}^n w_i - 2 \sum_{i=1}^n x'_i w_i \\ \frac{dE}{dt_y} = 0 &= 2s \sum_{i=1}^n y_i w_i + 2t_y \sum_{i=1}^n w_i - 2 \sum_{i=1}^n y'_i w_i \\ \frac{dE}{dt_z} = 0 &= 2s \sum_{i=1}^n z_i w_i + 2t_z \sum_{i=1}^n w_i - 2 \sum_{i=1}^n z'_i w_i \end{aligned}$$

Zur Vereinfachung seien:

$$\begin{aligned} \mathbf{S}_x &= \sum_{i=1}^n x_i w_i & \mathbf{S}_{x'} &= \sum_{i=1}^n x'_i w_i \\ \mathbf{S}_y &= \sum_{i=1}^n y_i w_i & \mathbf{S}_{y'} &= \sum_{i=1}^n y'_i w_i \\ \mathbf{S}_z &= \sum_{i=1}^n z_i w_i & \mathbf{S}_{z'} &= \sum_{i=1}^n z'_i w_i \\ \mathbf{S}_{xx+yy+zz} &= \sum_{i=1}^n (x_i^2 + y_i^2 + z_i^2) w_i & \mathbf{W} &= \sum_{i=1}^n w_i \\ \mathbf{S}_{xx'+yy'+zz'} &= \sum_{i=1}^n (x_i x'_i + y_i y'_i + z_i z'_i) w_i \end{aligned} \tag{A.7}$$

und damit

$$\begin{pmatrix} \mathbf{S}_{\mathbf{xx}+\mathbf{yy}+\mathbf{zz}} & \mathbf{S}_{\mathbf{x}} & \mathbf{S}_{\mathbf{y}} & \mathbf{S}_{\mathbf{z}} \\ \mathbf{S}_{\mathbf{x}} & \mathbf{W} & 0 & 0 \\ \mathbf{S}_{\mathbf{y}} & 0 & \mathbf{W} & 0 \\ \mathbf{S}_{\mathbf{z}} & 0 & 0 & \mathbf{W} \end{pmatrix} \begin{pmatrix} s \\ t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} \mathbf{S}_{\mathbf{xx}'+\mathbf{yy}'+\mathbf{zz}'} \\ \mathbf{S}_{\mathbf{x}'} \\ \mathbf{S}_{\mathbf{y}'} \\ \mathbf{S}_{\mathbf{z}'} \end{pmatrix}. \quad (\text{A.8})$$

Daraus kann man nun wieder ohne weiteres die optimalen Parameter  $s$ ,  $t_x$ ,  $t_y$  und  $t_z$  ermitteln.

## B Interpolation

Es gibt verschiedene Prozesse, bei denen die Textur eines Objektes mit dem Shape  $\mathbf{x}$  in das Shape  $\mathbf{x}'$  eines anderen Objektes projiziert werden muß. Dafür geht man davon aus, daß es eine Funktion  $\mathbf{f}$  gibt, die ein Shape in das andere überführt, also zu jedem Landmarkenpunkt  $\mathbf{x}_i$  angibt, wo sein Pendant  $\mathbf{x}'_i$  im Zielobjekt sich befindet. Die Anwendung von  $\mathbf{f}$  auf die inneren Punkte des Objektes ergibt dann eine Texturprojektion. In der Praxis führt dieses Verfahren zu störenden Löchern, weil z. B. das Zielobjekt mehr Punkte beinhaltet als das Ausgangsobjekt. Um dies zu vermeiden, sucht man die Umkehrfunktion  $\mathbf{f}'$  und kann damit für jeden Punkt im *Zielobjekt* berechnen, wo sich sein Gegenpart befindet.

Hat man die Ursprungskoordinaten eines Pixels oder Voxels errechnet, so liegt dieser selten exakt auf einem Punkt des diskreten Bildrasters. Also ist es ungenau, beispielsweise einfach den Grauwert des nächsten ganzzahligen Punktes zuzuweisen. Stattdessen bildet man den gewichteten Mittelwert derjenigen Pixel (4) bzw. Voxel (8), die Anteil an einem Pixel/Voxel der Größe 1 des Punktes mit stetigen Koordinaten haben.

Für das zweidimensionale Interpolationsproblem fiel die Wahl auf die von Bookstein [2] eingeführten **Thin Plate Splines**. Diese Methode hat den Vorteil, glatter zu sein als beispielsweise eine stückweise affine Interpolation. Beim affinen Verfahren werden die beiden Objekte anhand der gegebenen Randpunkte in äquivalente Dreiecke zu zerteilt. Jeder Punkt im jeweiligen Zieldreieck wird z. B. mit baryzentrischen Koordinaten [12] in Abhängigkeit der drei Eckpunkte dargestellt, wonach der Ausgangspunkt anhand der Eckpunkte des äquivalenten Ausgangsdreieck leicht zu ermitteln ist. Zwar sind Thin Plate Splines ein wenig kostenintensiver zu berechnen, aber das fällt bei den vorliegenden Größenordnungen kaum ins Gewicht.

Der dreidimensionale Fall ist weitaus komplexer. Verfahren, die eine elastische Anpassung der beiden Punktmengen ermöglichen, sind kompliziert und langwierig zu berechnen. Die Wahl fiel daher auf eine affine Transformation — das durch Umeyama

verbesserte Verfahren von Arun. Es liefert für zwei Punktmengen eine Rotationsmatrix, einen Translationsvektor und einen Skalierungsfaktor, mit denen sich die eine so auf die andere matchen läßt, daß der quadratische Fehler der Abstände korrespondierender Punkte minimiert wird. Das Verfahren läuft angenehm schnell ab, aber die Genauigkeit erschien hier nicht ausreichend, da die Suchfunktion sehr stark abhängig von der Interpolationsgüte ist. Ein Nachteil ist, daß sich zu viele Bereiche finden, in denen durch die Transformation Ursprungspunkte gefunden werden, die außerhalb der Ursprungstextur liegen. Die Zuordnung nächster Nachbarpunkte ergab an diesen Stellen störende Artefakte. Eine Verbesserung ließ sich durch folgendes Verfahren erreichen.

Die globale Transformation wurde durch mehrere lokale Transformationen ersetzt. Das Zielobjekt wurde dafür in Schichten zerlegt, 3-10 erwiesen sich als brauchbar. Die in einer solchen Schicht enthaltenen Kontrollpunkte des einen Shapes wurden affin mit den äquivalenten Punkten des anderen Shapes interpoliert. Die Texturvoxel wurden den selben Schichten zugeordnet und gemäß der entsprechenden Transformationsparameter interpoliert. Die Schichteinteilung wurde nach allen drei Koordinaten durchgeführt, wobei Überlappungsbereiche berücksichtigt wurden, um evtl. auftretende Artefakte an den Schichtgrenzen zu vermeiden. Der Mittelwert der drei Berechnungen wurde als Ergebnis genommen. Der Zeitfaktor für die reine Interpolation ist recht gering, für 20.000 Voxel genügen (an einem 2.6GHz-Rechner) wenige Sekunden. Deutlich aufwendiger ist die Bestimmung, ob ein Voxel in dem vom Shape umschlossenen Volumen liegt (Anhang C, S. 52).

## B.1 Zweidimensionale Interpolation mit Thin Plate Splines

### B.1.1 Eindimensionale Thin Plate Splines

Widmen wir uns zunächst dem einfachsten Fall. Sei  $U(a) = \left(\frac{a}{\sigma}\right)^2 \log\left(\frac{a}{\sigma}\right)$ , dabei  $\sigma$  ein Skalierungsfaktor der die Dicke des Splines definiert. Der eindimensionale Thin Plate Spline an einem Punkt  $x$  ist

$$t_{1d}(x) = \sum_{i=0}^{n-1} w_i U(|x - x_i|) + a_0 + a_1 x. \quad (\text{B.1})$$

Entscheidend ist die Wahl der Gewichte  $w_i$ ,  $a_0$  und  $a_1$ . Diese müssen so gewählt werden, daß sie die Bedingung  $\mathbf{f}(x_i) = x'_i$ ,  $i = 0 \dots n-1$  erfüllen. Betrachtet man alle  $n$  korrespondierenden Punktpaare, so ergibt das  $n$  lineare Gleichungen der Form

$$x'_j = \sum_{i=0}^{n-1} w_i U(|x_j - x_i|) + a_0 + a_1 x_j, \quad (\text{B.2})$$

bzw. in Matrix-/Vektorschreibweise mit  $U_{ij} = U_{ji} = U(|x_j - x_i|)$  folgendes Gleichungssystem:

$$\begin{pmatrix} U_{00} & \cdots & U_{0n-1} & 1 & x_0 \\ \vdots & \ddots & \vdots & 1 & \vdots \\ U_{n-10} & \cdots & U_{n-1n-1} & 1 & x_{n-1} \\ 1 & \cdots & 1 & 0 & 0 \\ x_0 & \cdots & x_{n-1} & 0 & 0 \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{n-1} \\ a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} x'_0 \\ \vdots \\ x'_{n-1} \\ 0 \\ 0 \end{pmatrix}. \quad (\text{B.3})$$

Durch Lösen dieses Systems lassen sich die Gewichte  $w_i$ ,  $a_0$  und  $a_1$  nun ohne weitere Schwierigkeiten berechnen.

### B.1.2 Mehrdimensionale Thin Plate Splines

Mehrdimensionale Thin Plate Splines funktionieren im Prinzip genauso wie eindimensionale. Aus einem eindimensionalen Punkt  $x_i$  wird ein  $d$ -dimensionaler Vektor  $\mathbf{x}_i$  und damit

$$U(|\mathbf{x} - \mathbf{x}_i|) = U\left(\sqrt{(x_0 - x_{i0})^2 + \cdots + (x_{d-1} - x_{id-1})^2}\right). \quad (\text{B.4})$$

Das zu lösende lineare Gleichungssystem verkompliziert sich ein wenig zu:

$$AW = X', \quad (\text{B.5})$$

mit

$$A = \begin{pmatrix} U_{00} & \cdots & U_{0n-1} & 1 & x_{00} & \cdots & x_{0d-1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ U_{n-10} & \cdots & U_{n-1n-1} & 1 & x_{n-10} & \cdots & x_{n-1d-1} \\ 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ x_{00} & \cdots & x_{n-10} & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{0d-1} & \cdots & x_{n-1d-1} & 0 & 0 & \cdots & 0 \end{pmatrix} \quad (\text{B.6})$$

sowie

$$W = \begin{pmatrix} w_{00} & \cdots & w_{d-10} \\ \vdots & \ddots & \vdots \\ w_{0n-1} & \cdots & w_{d-1n-1} \\ a_{00} & \cdots & a_{d-10} \\ \vdots & \ddots & \vdots \\ a_{0d} & \cdots & a_{d-1d} \end{pmatrix}, \quad X' = \begin{pmatrix} x'_{00} & \cdots & x'_{d-10} \\ \vdots & \ddots & \vdots \\ x'_{0n-1} & \cdots & x'_{d-1n-1} \\ 0_{00} & \cdots & 0_{d-10} \\ \vdots & \ddots & \vdots \\ 0_{0d} & \cdots & 0_{d-1d} \end{pmatrix}. \quad (\text{B.7})$$

## B.2 Dreidimensionale affine Interpolation

Der Algorithmus von Arun beschreibt die Shapetransformation vermittels einer Rotationsmatrix, einem Translationsvektor und einem Skalierungsfaktor. Diese Parameter werden aufgrund der Minimierung des quadratischen Fehlers ermittelt. Das Problem, daß bei starkem Rauscheinfluß zuweilen statt einer Rotation eine Spiegelung berechnet wird, löste Umeyama. Der so verbesserte Algorithmus [24] wurde für diese Arbeit implementiert.

Gegeben sind wieder zwei Shapes  $\mathbf{x}$  und  $\mathbf{x}'$  mit jeweils  $N$  paarweise korrespondierenden Punkten. Die affine Beziehung zwischen ihnen lautet für jeden Punkt  $i$

$$\mathbf{x}'_i = cR\mathbf{x}_i + \mathbf{t} + \mathbf{E}_i, \quad (\text{B.8})$$

wobei  $c$  der Skalierungsfaktor,  $R$  die  $3 \times 3$  Rotationsmatrix,  $\mathbf{t}$  der Translationsvektor und  $\mathbf{E}_i$  ein Restterm sind, da ein exaktes Matching so nicht möglich ist. Nach  $\mathbf{E}_i$  umgestellt, quadriert und über alle Punkte aufsummiert, ergibt sich auf der rechten

Seite die zu minimierende Summe

$$S^2 = \frac{1}{N} \sum_{i=0}^{N-1} \| \mathbf{x}'_i - (c\mathbf{R}\mathbf{x}_i - \mathbf{t}) \|^2. \quad (\text{B.9})$$

Seien nun

$$\mu_x = (\mu_{x0}, \mu_{x1}, \mu_{x2})^T = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{x}_i \quad \text{und} \quad (\text{B.10})$$

$$\mu_{x'} = (\mu_{x'0}, \mu_{x'1}, \mu_{x'2})^T = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{x}'_i \quad (\text{B.11})$$

die Schwerpunkte der beiden Shapes. Eine Lösung des Minimierungsproblems hat ihren Schwerpunkt in  $\mu_{x'}$ . Damit läßt sich, gegeben  $c$  und  $\mathbf{R}$ , die Translation  $\mathbf{t}$  nach

$$\mathbf{t} = \mu_{x'} - c\mathbf{R}\mu_x \quad (\text{B.12})$$

berechnen. Zentriert man beide Shapes im Koordinatenursprung, so reduziert sich das Problem um den Translationsvektor  $\mathbf{t}$ . Daher werden nun

$$\mathbf{q}_{x_i} = (q_{x_{i0}}, q_{x_{i1}}, q_{x_{i2}})^T = \mathbf{x}_i - \mathbf{t} \quad \text{und} \quad (\text{B.13})$$

$$\mathbf{q}_{x'_i} = (q_{x'_{i0}}, q_{x'_{i1}}, q_{x'_{i2}})^T = \mathbf{x}'_i - \mathbf{t} \quad (\text{B.14})$$

gesetzt. Seien nun

$$\sigma_x^2 = \frac{1}{N} \sum_{i=0}^{N-1} \| \mathbf{q}_i \|^2 \quad \text{und} \quad (\text{B.15})$$

$$\mathbf{H} = \frac{1}{N} \sum_{i=0}^{N-1} (\mathbf{q}_{x'_i})(\mathbf{q}_{x_i})^T \quad (\text{B.16})$$

Varianz der Abstände in  $\mathbf{x}$  und  $3 \times 3$  Kovarianzmatrix beider Shapes, so sind nach einer Singulärwertzerlegung von  $\mathbf{H}$  in  $\mathbf{H} = \mathbf{USV}^T$  die beiden verbleibenden Parameter gemäß

$$\mathbf{R} = \mathbf{US}_1\mathbf{V}^T \quad \text{und} \quad (\text{B.17})$$

$$c = \frac{1}{\sigma_x^2} \text{spur}(\mathbf{SS}_2) \quad (\text{B.18})$$



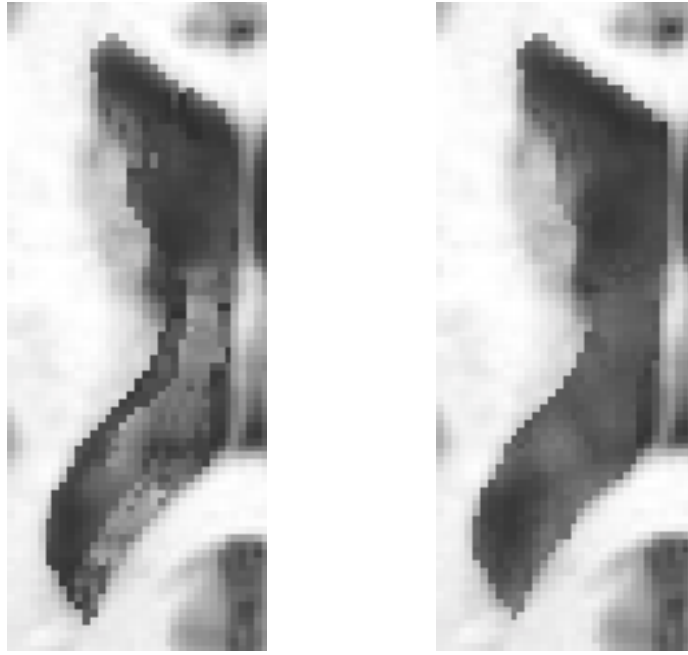


Abbildung B.1: Darstellung einer Schicht beim Interpolationsvergleich — Verfahren nach Arun und Umeyama — Die Textur eines 3D-Objektes wurde in ein anderes projiziert: globale Anwendung links, lokale Anwendung rechts

---

zu berechnen. Dabei müssen die Singulärwerte in  $\mathbf{S}$  absteigend sortiert sein und die Matrizen  $\mathbf{S}_1$  und  $\mathbf{S}_2$  ergeben sich mit

$$\mathbf{S}_1 = \begin{cases} \mathbf{I}_3 & \text{wenn } \det(\mathbf{H}) \geq 0 \\ \text{diag}(1, 1, -1) & \text{sonst} \end{cases} \quad (\text{B.19})$$

$$\mathbf{S}_2 = \begin{cases} \mathbf{I}_3 & \text{wenn } \det(\mathbf{U}) \det(\mathbf{V}) = 1 \\ \text{diag}(1, 1, -1) & \text{wenn } \det(\mathbf{U}) \det(\mathbf{V}) = -1 \end{cases} \quad (\text{B.20})$$

Mit den Gleichungen (B.12), (B.17) und (B.18) sind also die optimalen Transformationsparameter gefunden.

Anhand einer 3D-Ventrikelerpolation sind in Abbildung B.1 die zu z. T. deutlichen Unterschiede zwischen der Interpolation mit dem vollen Datensatz und der schichtweisen Variante (hier 5 Schichten, Überlappungszonen von 6 Pixeln) verdeutlicht.

## C Volumenbestimmung

Für die Interpolationsprozesse ist es Voraussetzung, daß bekannt ist, welche Pixel bzw. Voxel sich innerhalb des Objektes befinden, in das die Textur projiziert werden soll. Im zweidimensionalen Raum ist ein Volumen eine Fläche. Diese wird durch die Strecken zwischen je zwei benachbarten Kontrollpunkten definiert. Im dreidimensionalen Raum wird das Volumen durch ein geschlossenes Gitter von Dreiecksflächen begrenzt. Die Zuweisung der Punkte geschah manuell — ein sehr aufwendiger Pozeß, aber einmal durchgeführt, gilt die gewonnene Triangulierung für alle Objekte dieser Klasse, da sich die Kontrollpunkte an äquivalenten Stellen befinden. Die Frage, wie sich entscheiden läßt, ob ein Punkt innerhalb oder außerhalb des Volumens liegt, ist durch einen prinzipiell sehr einfachen Gedankengang zu beantworten.

Betrachten wir zunächst den zweidimensionalen Fall. Gegeben sind der Punkt  $P = (x_P, y_P)$  und eine Menge von Randstrecken. Die Fläche sei zunächst konvex. Bewegt man sich von  $P$  geradlinig in eine definierte Richtung beispielsweise in Richtung der  $x$ -Koordinate (nach rechts), so tritt einer der folgenden Fälle auf. Es gibt Ausnahmen, aber dazu später. Liegt  $P$  innerhalb der Fläche, dann muß zwangsläufig irgendwann der Rand durchbrochen werden. In diesem Fall existiert ein Schnittpunkt des von  $P$  ausgehenden Strahls mit einer der Randstrecken. Liegt  $P$  dagegen außerhalb, so kann er sich rechts befinden, woraufhin man keinen Schnittpunkt fände oder links, was zwei Schnittpunkte ergäbe. Ist die Fläche nicht konvex, so ändern sich die Möglichkeiten derart, daß der Strahl für innerhalb gelegene Punkte  $P$  eine *ungerade*, ein außerhalb gelegene eine *gerade* Anzahl von Schnittpunkten mit den Strecken erfährt (Abb. C.1(a)).

Es gibt Ausnahmen von dieser Regel, wie in Abb. C.1(b) dargestellt. Wenn solch ein Strahl eine waagerechte Strecke oder einen Kontrollpunkt, der gleichzeitig ein zumindest lokales Maximum trifft, so reduziert sich die Anzahl der Schnittpunkte um einen. Wird dagegen ein „normaler“ Kontrollpunkt getroffen, so erhöht sich die Anzahl um eins, weil anstatt einer zwei Strecken getroffen werden. Beides läßt sich

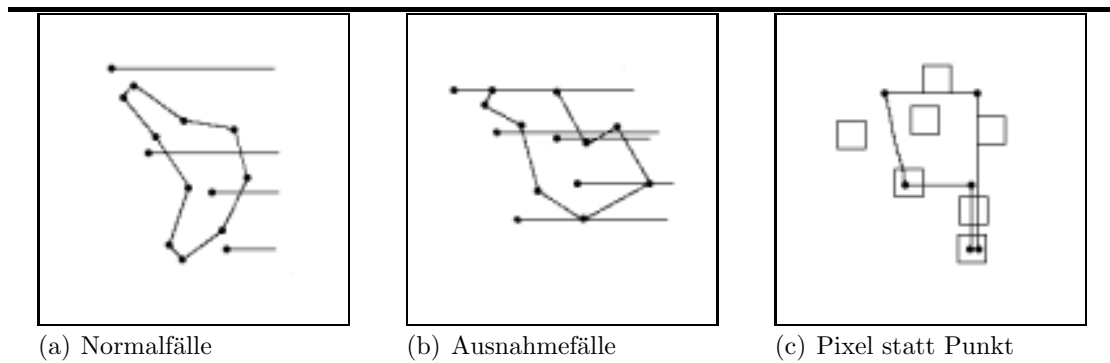


Abbildung C.1: Volumenbestimmung in 2D

beheben, indem man ein klein wenig über- oder unterhalb des Punktes noch einmal testet. Wichtig ist es nun, zu beachten, daß ein Pixel kein Punkt, sondern ein Viereck ist. Ein Pixel liegt also genau dann innerhalb der Fläche, wenn sich die Fläche mit der Pixelfläche irgendwo überdeckt, im Normalfall also, wenn sich mindestens einer der Pixeleckpunkte innerhalb der Fläche befindet. Bilden zwei Strecken aber einen sehr spitzen Winkel oder verlaufen (fast) parallel und haben dabei einen Abstand kleiner als 1, so kann es vorkommen, daß sie das Pixel durchstoßen, ohne daß einer der Eckpunkte berührt wird. Darum wird ggf. auch noch getestet, ob eine Strecke eine der Pixelrandkanten schneidet. Diese Problematik ist in Abb. C.1(c) verdeutlicht.

Im dreidimensionalen Raum funktioniert das Verfahren analog. Es dauert allerdings viel länger. Für jeden zu testenden Punkt  $P = (x_P, y_P, z_P)$  muß jedes Dreieck, wenn  $y$ - und  $z$ -Koordinate einen möglichen Schnittpunkt nahelegen, dahingehend untersucht werden. Dazu wird die Hessesche Normalform der Ebenengleichung des Dreiecks aufgestellt, daraus der Normalenvektor, also der Vektor, der senkrecht zur Ebene verläuft, abgelesen und damit der Lotfußpunkt von  $P$  auf die Ebene ermittelt. Für diesen werden dann sphärische Baryzentrische Koordinaten [18] errechnet. Sind diese alle größer oder gleich 0, so liegt der Lotfußpunkt innerhalb der Dreiecksfläche. In diesem Fall ist dann an jener Stelle der Rand durchbrochen worden. Aus einer ungeraden Anzahl solcher Schnittpunkte ergibt sich, daß ein Punkt sich innerhalb des Volumens befindet. Die für den zweidimensionalen Fall genannten Ausnahmen gelten hier in erweiterter Form. Trifft ein Punkt direkt auf eine Dreiecksseite oder einen Kontrollpunkt, so muß wieder etwas darüber oder darunter noch einmal gete-

stet werden. Dahingehend problematisch sind senkrechte Kanten, so daß mit kleiner  $z$ - und  $y$ -Abweichung je noch einmal getestet wird. Für direkte Treffer auf Kontrollpunkte muß noch genauer gesucht werden, da in diesen senkrechte wie waagerechte Kanten aufeinandertreffen können.

Auch hier werden keine einzelnen Punkte betrachtet, sondern Raumwürfel, sogenannte Voxel. Wieder ist ein Voxel Teil des Volumens, wenn sich ein Teil von ihm mit einem Teil des Shape-Volumens deckt. Im Normalfall tritt dies ein, sobald der Test für einen der Eckpunkte positiv verläuft. Wie bei Pixeln können auch hier durch sehr schmale Strukturen Voxel durchstoßen werden, weswegen auch noch die Kanten eines Dreiecks auf Schnittpunkte mit den Voxelflächen untersucht werden müssen.

Für die untersuchten 3D-Objekte erwies sich dieses Verfahren in der Interpolation als der zeitlich bestimmende Faktor. Selbst die aufgrund der Schichtzerlegung vielfach durchgeführte Ermittlung der affinen Transformationsparameter und wegen der Überlappungszonen etwas mehr als dreifachen Anwendung auf die Voxel verlief sehr viel schneller als die vorangegangene Bestimmung des Volumens. Ein schnellerer Algorithmus wäre der praktischen Anwendung sehr dienlich.

## D Beschleunigung der Hauptkomponentenanalyse

Die Hauptkomponentenanalyse (PCA) während des Trainingsprozesses kann für große Modelle sehr schnell sehr aufwendig werden, denn ihr Rechenaufwand steigt kubisch mit der Länge der beteiligten Vektoren, also der Anzahl der Modellpunkte oder der Grauwerte an. Insbesondere letzteres trifft bei dreidimensionalen Modellen zu — die schiere Anzahl der Grauwerte macht eine PCA in der Praxis fast unmöglich. In [7] geben T. F. Cootes und C. J. Taylor eine Methode an, wie man PCAs sparsamer durchführen kann, wenn weniger Vektoren als Dimensionen vorhanden sind. Ziel der PCA ist es, die Eigenvektoren der Kovarianzmatrix  $\mathbf{S}$  zu ermitteln und dadurch eine Orthonormalbasis des  $n \cdot \dim$ -Raumes zu finden. Gerade der Prozeß der Eigenwert- und Eigenvektorfindung ist es aber, der die Kosten derart in die Höhe treibt. Die Kovarianzmatrix von  $s$  Vektoren der Länge  $n$  berechnet sich nach

$$\mathbf{S} = \frac{1}{s} \mathbf{D} \mathbf{D}^T, \quad (\text{D.1})$$

mit

$$\mathbf{D} = ((\mathbf{x}_0 - \bar{\mathbf{x}}) \mid \dots \mid (\mathbf{x}_{n-1} - \bar{\mathbf{x}})). \quad (\text{D.2})$$

Berechnet man stattdessen die Eigenvektoren der Matrix

$$\mathbf{T} = \frac{1}{s} \mathbf{D}^T \mathbf{D}, \quad (\text{D.3})$$

so sind zunächst alle Eigenwerte  $\mathbf{e}_i$  von  $\mathbf{T}$  auch Eigenwerte von  $\mathbf{S}$ , alle anderen Eigenwerte von  $\mathbf{S}$  sind 0. Die Vektoren  $\mathbf{b}_i = \mathbf{D} \mathbf{e}_i$  bilden zunächst eine Basis des  $n \cdot \dim$ -Raumes, sind aber nicht automatisch orthonormal, nicht einmal orthogonal. Eine Orthonormalisierung nach Gram/Schmidt wandelt die Basis entsprechend.

Dieses Verfahren ist ein rekursiver Prozeß und läuft nach folgendem Muster ab:

$$\begin{aligned} \mathbf{u}_0 &= \frac{\mathbf{b}_0}{\|\mathbf{b}_0\|} & , & & \mathbf{v}_i &= \mathbf{b}_i - \sum_{k=0}^{i-1} (\mathbf{u}_k \cdot \mathbf{b}_i) \mathbf{u}_k & \quad | i > 0 \\ \mathbf{u}_i &= \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} & \quad | i > 0 \end{aligned}$$

Die Vektoren  $\mathbf{u}_i$  ( $i \geq 0$ ) bilden nun eine Orthonormalbasis des  $n \cdot \dim$ -Raumes.

# Literaturverzeichnis

- [1] M. D. Akca, *Generalized Procrustes analysis and its applications in photogrammetry*, S. 4-15, ETH, Swiss Federal Institute of Technology Zurich, Institute of Geodesy and Photogrammetry (<http://e-collection.ethbib.ethz.ch/show?type=bericht&nr=363>), 2003
- [2] F. L. Bookstein, *Principal warps: Thin-plate splines and the decomposition of deformations*, S. 567-585, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989
- [3] L. D. Cohen, *On Active Contour Models and Ballons*, *Image Understanding*, 1991
- [4] T. F. Cootes, T. F. Taylor, C. J. Cooper, J. Graham, *Active shape models — their training and application*, S. 38-59, *Computer Vision and Image Understanding*, 61(1), 1995
- [5] T. F. Cootes, G. Edwards, C. J. Taylor, *A Comparative Evaluation of Active Appearance Model Algorithms*, Proc. British Machine Vision Conference, 1998, (Eds: P.H. Lewis and M.S. Nixon) Vol. 2, S. 680-689, BMVA Press, 1998
- [6] T. F. Cootes, G. J. Edwards, C. J. Taylor, *Active Appearance Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol.23, No.6, S. 683, Juni 2001
- [7] T. F. Cootes, C. J. Taylor, *Statistical models of appearance for medical image analysis and computer vision*, Proc. SPIE Medical Imaging, 2001
- [8] T. F. Cootes, C. J. Taylor, *Statistical Models of Appearance for Computer Vision*, 2001

- [9] J. E. Desmond, K. O. Lim, *On- and offline Talairach registration for structural and functional MRI studies*, Human Brain Mapping 1, S. 58 - 73, 1997
- [10] L. Dornheim, *Generierung und Dynamik physikalisch basierter 3D-Modelle zur Segmentierung des linken Ventrikels in SPECT-Daten*, Diplomarbeit an der Otto-von-Guericke-Universität Magdeburg, Januar 2005
- [11] A. D. Elster, J. H. Burdette, *Questions & Answers in Magnetic Resonance Imaging*, S. 19-101, Mosby-Verlag, 2001
- [12] G. Farin, *Kurven und Flächen im Computer Aided Geometric Design*, S. 247-249, Vieweg 1994
- [13] M. Kass, A. Witkin, D. Terzopoulos, *Snakes: Active contour models*, S. 321-331, International Journal of Computer Vision, 8(2)
- [14] A. Kelemen, G. Székely and G. Gerig, *Three-dimensional Model-based Segmentation*, Image Science Lab, ETH Zürich, 1997
- [15] V. D. Köchli, B. Marincek, *Wie funktioniert MRI? Eine Einführung in Physik und Funktionsweise der Magnetresonanzbildung*, S. 1-26, 47-51, Springer Verlag, 1998
- [16] D. Maier, J. Hessler, R. Männer, *Fast and Accurate Closest Point Search on Triangulated Surfaces and its Application to Head Motion Estimation*, In 3rd WSEAS International Conference on SIGNAL, SPEECH and IMAGE PROCESSING, Oktober 2003
- [17] S. de Pablo, S. Lorenzo, J. I. Herrero, *Alternating Versions of Snakes*, SNRFAI 1997, VII Simposium Nacional de Reconocimiento de Formas y Análisis de Imagen, Barcelona, 1997
- [18] R. Pfeifle, H. Seidel, *Spherical Triangular B-Splines with Applications to Data Fitting*, S. C-90, Eurographics, 1995
- [19] E. J. Rummeny, P. Reimer, W. Heindel, *Ganzkörper MR-Tomographie*, S. 2-23, Thieme-Verlag, 2002
- [20] S. Sclaroff, J. Isidoro, *Active Blobs*, 6<sup>th</sup> International Conference on Computer Vision, 1998



- [21] L. I. Smith, *A Tutorial on PCA*, Maintained by Cornell University, USA, April 2002
- [22] M. B. Stegmannm, R. Fisker, B. K. Ersbøll, H. H. Thodberg, L. Hyldstrup, *Active Appearance Models: Theory and Cases*, 2000
- [23] S. Rusinkiewicz, M. Levoy, *Efficient Variants of the ICP Algorithm*, Third International Conference on 3D Digital Imaging and Modeling (3DIM 2001)
- [24] Katrin Wedekind, *Analyse, Implementierung und Evaluation von Matching-verfahren in der medizinischen Bildverarbeitung*, Technical Report No. 79/1995, <http://mbi.dkfz-heidelberg.de/mbi/TR/TR79/index.html>
- [25] A. L. Yuille, D. S. Cohen, P. Hallinan, *Feature extraction from faces using deformable templates*, International Journal of Computer Vision, 8(2): 99-112, 1992

# Erklärung

Ich versichere, daß ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig, den 6. Oktober 2005